BAAN IV

Conversion TRITON 3 to BAAN IV

Conversion Guide U7025C US

Research and Development

Document information

Document

Number : U7025C US
Group : User documentation
Name : Conversion TRITON 3 to BAAN IV
Edition : C
Date : July 1997

© 1997 Baan Development B.V. All rights reserved.

The information in this document is subject to change without notice. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Baan Development B.V.

Baan Development B.V. assumes no liability for any damages incurred, directly or indirectly, from any errors, omissions or discrepancies between the software and the information contained in this document.

Table of contents

1	**Introduction**	1.1
1.1	Scope	1.1
1.2	Target group	1.1
1.3	Remarks and recommendations	1.1
	1.3.1 Conversion to BAAN IV: two possible scenarios	1.1
	1.3.2 Customizations	1.3
	1.3.3 Localizations	1.3
	1.3.4 Disk space	1.3
	1.3.5 Straight migration	1.4
	1.3.6 Test conversion	1.4
	1.3.7 Integration of implementation and data conversion	1.4
1.4	Terms and abbreviations	1.5
1.5	References	1.5
2	**Preparing TRITON 3 for conversion: the two environments scenario**	2.1
2.1	Steps in the procedure	2.1
2.2	General	2.3
	2.2.1 Checking the consistency of the TRITON 3 data	2.3
	2.2.2 Create a backup of the TRITON software and data	2.3
	2.2.3 Export Users/Authorizations/Devices/Currencies/Logical Tables (ottcon61export)	2.4
2.3	Software export	2.5
	2.3.1 Export Data Dictionary (ttiex1280m000)	2.5
2.4	Data export	2.5
	2.4.1 General data export	2.6
	2.4.2 Database-specific data transfer	2.6
3	**Converting to BAAN IV: two environments scenario**	3.1
3.1	Steps in the procedure	3.1
3.2	General	3.6
	3.2.1 Import Users/Authorizations/Devices/Currencies/Logical Tables (ttcon61import)	3.6
3.3	Software conversion	3.7
	3.3.1 Import Data Dictionary (ttiex1285m000)	3.7
	3.3.2 Maintain Packages by Package Combination (ttaad1121m000)	3.7
3.4	Data conversion	3.8
	3.4.1 Copy Package Combination (ttaad1220m000)	3.8
	3.4.2 Copy data definition files	3.8
	3.4.3 Maintain Companies (ttaad1100m000)	3.9
	3.4.4 Assign Tables to Databases (ttaad4111m000)	3.9
	3.4.5 Maintain Table Directories (BAAN Base (TP)) (ttaad4115m000)	3.9
	3.4.6 Maintain Logical Tables (ttaad4120m000)	3.9
	3.4.7 Maintain User Data (ttaad2100m000)	3.10
	3.4.8 Data import	3.10
	3.4.9 Check Tables (ttaad4232m000)	3.11
	3.4.10 Convert 3.0/3.1 Tables to BAAN IV (ttaad1102m000)	3.11
	3.4.11 Check Tables (ttaad4232m000)	3.11
	3.4.12 Change Package Combination for Users (ttaad2200m000)	3.11

3.4.13 Convert User Data to Runtime Data Dictionary (ttaad4200m000) 3.11
3.4.14 Maintain Package Combinations (ttaad1120m000) 3.12
3.4.15 Create Tables (ttaad4230m000) 3.12
3.4.16 Reorganize Tables (ttaad4225m000) 3.12

4 Converting to BAAN IV: the single environment scenario 4.1
4.1 Steps in the procedure 4.1
4.2 Preparation 4.5
4.2.1 Backup 4.5
4.2.2 TRITON 4.5
4.2.3 Time frame 4.5
4.2.4 Prerequisites 4.5
4.2.5 Test 4.5
4.2.6 Customizations 4.6
4.2.7 Notes 4.6
4.3 Installation procedure 4.7
4.3.1 Step 1: Reading out tapes 4.7
4.3.2 Step 2: Start installation script 4.7
4.3.3 Step 3: Running preinst during or right after the installation of BAAN IV 4.8
4.3.4 Step 4: Check installation logs 4.9
4.3.5 Step 5: Import the VRC containing the software components for conversion 4.10
4.3.6 Step 6: B/C - Import customization (ttiex1285m000) 4.10
4.3.7 Step 7: Create new VRCs (ttadv1111m000) 4.10
4.3.8 Step 8: Create directories for new VRCs (ttadv1115m000) 4.11
4.3.9 Step 9: (Own) Copying own work to new own VRCs (ttadv6265m000) 4.11
4.3.10 Step 10: Create New Package Combination(s) (ttaad1120m000) 4.11
4.3.11 Step 11: Fill Package Combination(s) with VRCs (ttaad1221m000) 4.12
4.3.12 Step 12: Migrate company/companies (ttaad1101m000) 4.12
4.3.13 Step 13: Migrate user(s) (ttaad2200m000) 4.12
4.3.14 Step 14: (Own) Dump forms and menus (ttadv5214m000) 4.12
4.3.15 Step 15: (Own) Compile reports (ttadv3240m000) 4.13
4.3.16 Step 16: Validate the system 4.13
4.3.17 Step 17: Create the audit database and assign tables to it 4.14
4.3.18 Step 18: Create tables (ttaad4230m000) 4.15
4.3.19 Step 19: Parameters (tcmcs0295m000) 4.15
4.3.20 Step 20: Maintain Company Parameters (tfgld0103m000) 4.15
4.3.21 Step 21: Maintain ITM Parameters (tiitm0100m000) 4.15
4.3.22 Step 22: Maintain Base Units (tcmcs0100m000) 4.15
4.3.23 Step 23: Checking and finalizing 4.15

5 Correction program and checking the conversion: both scenarios 5.1
5.1 Conversion matrix release 3 release 4 5.1
5.2 Conversion of TRITON 3.x => BAAN IV (tccov0000m000) 5.3
5.2.1 BAAN Distribution 5.3
5.2.2 BAAN Manufacturing 5.8
5.2.3 BAAN Finance 5.10
5.2.4 BAAN Transportation 5.11
5.2.5 Converting TRITON Service version 3.0b to BAAN IV 5.11
5.3 General data check 5.14
5.3.1 TRITON Common 5.14
5.3.2 TRITON Distribution 5.14

5.3.3 TRITON Manufacturing 5.15
5.3.4 TRITON Finance 5.16
5.4 Specific data check 5.16
5.4.1 BAAN Distribution 5.16
5.4.2 BAAN Manufacturing 5.17
5.4.3 BAAN Finance 5.17
5.4.4 BAAN Manufacturing 5.17
5.5 Conversion BAAN IVa to BAAN IVb 5.18
5.5.1 BAAN Tools 5.18
5.5.2 BAAN Enterprise Modeler 5.18
5.5.3 BAAN Enterprise Performance Manager 5.18
5.5.4 Data conversion from BAAN IVa to BAAN IVb 5.18
5.6 Converting localizations to BAAN IV 5.19
5.6.1 Proforma Invoice Conversion from 3.1x to BAAN IVx (tfcmgl900m000) 5.19
5.6.2 EDI 'Message Generated' Conversion from BAAN IVa to BAAN IVb (tccovl999m000) 5.19
5.7 Removing the 3.*x* software after the conversion 5.19

6 Appendix A: Upgrading customizations 6.1
6.1 Recommendations 6.1
6.2 Print differences between two PVRCs 6.1
6.3 Upgrade of in-house customizations 6.1
6.4 Upgrading customized software and in-house customizations 6.3

7 Appendix B: Troubleshooting 7.1
7.1 Conversion of Distribution 3.1 to BAAN IV (tccov1001m000) 7.1
7.1.1 Message 1: START OF CONVERSION tccov1001m000 by user <logon name> 7.1
7.1.2 Message 2: END OF CONVERSION tccov1001m000 by user <logon name> 7.1
7.1.3 Message 3: 'Letter layout: <tdsmi150.clet> converted' 7.1
7.1.4 Message 4: 'Selection expression: <tdsmi070.cnsc> converted' 7.1
7.1.5 Message 5: 'Recommended Purchase Order: <tdinv310.orno> converted' 7.1
7.1.6 Message 6: 'Outgoing Message: <tcedi700.mess> converted' 7.2
7.1.7 Message 7: 'Batch Number: <tcedi720.bano> converted' 7.2
7.1.8 Message 8: 'Company Purchase Order: <tdpur045.orno> <tdpur045.pono> <tdpur045.srnb>converted' 7.2
7.1.9 Message 9: 'Financial Customer Group Sales Order: <tdsls040.orno> converted' 7.2
7.2 Conversion of Distribution 3.1 to BAAN IV (tccov1002m000) 7.3
7.2.1 Message 1: START OF CONVERSION tccov1002m000 by user <logon name> 7.3
7.2.2 Message 2: END OF CONVERSION tccov1002m000 by user <logon name> 7.3
7.2.3 Message 3: 'Conversion error: sst010.sstr:<sst010.sstr>' 7.3
7.2.4 Message 4: 'Conversion error: sst015.sstr: <sst015.sstr>' 7.4
7.2.5 Message 5: 'Conversion error: sst018.sstr: <sst018.sstr>' 7.4
7.2.6 Message 6: 'Conversion error: sst020.sstr: <sst020.str>' 7.4
7.2.7 Message 7: 'Conversion error: sst025.sstr: <sst025.str>' 7.4
7.2.8 Message 8: 'Conversion error: sst028.sstr: <sst028.str>' 7.4
7.2.9 Message 9: 'Conversion error: sst050.sstr: <sst050.str>' 7.5
7.2.10 Message 10: 'Conversion error: sst030.sstr: <sst030.str>' 7.5
7.2.11 Message 11: 'Conversion error: pst010.sstr:<pst010.sstr>' 7.5

7.2.12 Message 12: 'Conversion error: pst015.sstr:<pst015.sstr>' 7.5
7.2.13 Message 13: 'Conversion error: pst018.sstr:<pst018.sstr>' 7.5
7.2.14 Message 14: 'Conversion error: pst020.sstr:<pst020.sstr>' 7.6
7.2.15 Message 15: 'Conversion error: pst025.sstr:<pst025.sstr>' 7.6
7.2.16 Message 16: 'Conversion error: pst028.sstr:<pst028.sstr>' 7.6
7.2.17 Message 17: 'Conversion error: pst050.sstr:<pst050.sstr>' 7.6
7.2.18 Message 18: 'Conversion error: pst030.sstr:<pst050.sstr>' 7.6
7.2.19 Message 19: 'Conversion error: mcs047.ckon: <mcs047.ckon> mcs047.grno:<mcs047.grno>' *7.7*
7.2.20 Message 20: 'Conversion error: edi005.mess <tcedi005.mess>' *7.7*
7.2.21 Message 21: 'Conversion error: pur002.opol <pur002.qono> <pur002.pono>' 7.7
7.2.22 Message 22: 'Conversion error: pur011.opol <pur011.qono> <pur011.pono>' 7.7
7.2.23 Message 23: 'Conversion error: pur034.suno <pur034.suno> pur034.cpgp <pur034.cpgp>pur034.dile <pur034.dile>' 7.8
7.2.24 Message 24: 'Conversion error: pur035.cplp <pur035.cplp> pur035.cpgp <pur035.cpgp>pur035.dile <pur035.dile>' 7.8
7.2.25 Message 25: 'Conversion error: pur036.suno <pur036.suno> pur036.item <pur036.item>pur036.cntr <pur036.cntr> pur036.dile <pur036.dile>' 7.8
7.2.26 Message 26: 'Conversion error: pur037.cplp <pur037.cplp> pur037.item <pur037.item>pur037.cntr <pur037.cntr> pur037.dile <pur037.dile>' 7.8
7.2.27 Message 27: 'Conversion error: inv300.orno <tdinv300.orno>' 7.8
7.2.28 Message 28: 'Conversion error: inv310.orno <tdinv310.orno>' 7.9
7.2.29 Message 29: 'Conversion error: sls002.opol <sls002.qono> <sls002.pono>' 7.9
7.2.30 Message 30: 'Conversion error: sls011.opol <sls011.qono> <sls011.pono>' 7.9
7.2.31 Message 31: 'Conversion error: sls034.cuno <sls034.cuno> sls034.cpgp <sls034.cpgp>sls034.dile <sls034.dile>' 7.9
7.2.32 Message 32: 'Conversion error: sls035.cpls <sls035.cpls> sls035.cpgp <sls035.cpgp>sls035.dile <sls035.dile>' 7.9
7.2.33 Message 33: 'Conversion error: sls036.cuno <sls036.cuno> sls036.item <sls036.item>sls036.cntr <sls036.cntr> sls036.dile <sls036.dile>' 7.10
7.2.34 Message 34: 'Conversion error: sls037.cpls <sls037.cpls> sls037.item <sls037.item>sls037.cntr <sls037.cntr> sls037.dile <sls037.dile>' 7.10
7.2.35 Message 35: 'Conversion error: sls045.orno <sls045.orno> sls045.pono <sls045.pono>sls045.srnb <sls045.srnb>' 7.10
7.2.36 Message 36: 'Conversion error: sls051.orno <sls051.orno> sls051.trdt <sls051.trdt>sls051.trtm <sls051.trtm>' 7.10
7.2.37 Message 37: 'Conversion error: com999.sern <tccom999.sern>' 7.10
7.2.38 Message 38: 'Conversion error: mcs042.blcs: blocking stepno not in procedure' 7.11
7.2.39 Message 39: 'Conversion error: tcmcs042.pmsk: <tcmcs042.cotp> <tcmcs042.pmsk>' *7.11*
7.2.40 Message 40: 'Conversion error: tdrpl004.pmsk: <tdrpl004.cotp> <tdrpl004.pmsk>' *7.11*
7.2.41 Message 41: 'Conversion error: tdpur045.pmsk: <tdpur045.orno> <tdpur045.pono><tdpur045.srnb> <tdpur045.pmsk>' 7.11
7.2.42 Message 42: 'Conversion error: tdrpl100.pmsk: <tdrpl100.orno> <tdrpl100.pono><tdrpl100.srnb> <tdrpl100.pmsk>' 7.12
7.3 Changing a sign in open item table (tccov2001m000) 7.12

7.3.1 Message 1: START OF CONVERSION tccov2001m000 by user <login name> 7.12
7.3.2 Message 2: Convert CDAM error: <supplier>,<transactiontype>,<document>,<linenumber>, <transactiontype>,<document>,<linenumber>,<currency difference> 7.12
7.3.3 Message 3: END OF CONVERSION 7.12
7.4 Change step for anticipated unallocated/advance payments (tccov2002m000) 7.13
7.4.1 Message 1: START OF CONVERSION tccov2002m000 by user <login name> 7.13
7.4.2 Message 2: tfacp200: <supplier>,<transactiontype>,<document>,<linenumber>,\<transactiontype> ,<document>,<linenumber>\ 7.13
7.4.3 Message 3: tfacr200: <customer>,<transactiontype>,<document>,<linenumber>,<transactiontype >,<document>, <linenumber> 7.13
7.4.4 Message 4: END OF CONVERSION 7.13
7.5 tccov2003m000: Checking standing order (tfcmg114) 7.14
7.5.1 Message 1: START OF CONVERSION tccov2003m000 by user <login name> 7.14
7.5.2 Message 2: Posting Data: <bank>,<document>,<company>, <transactiontype>,<document>,<supplier>, <standing order>,<serial number> 7.14
7.5.3 END OF CONVERSION 7.14
7.6 tccov2004m000: Changing anticipated documents in tfacp200/tfacr200/tfgld102/tfgld106 7.15
7.6.1 Message 1: START OF CONVERSION tccov2004m000 by user <login name> 7.15
7.6.2 Message 2: tfacp200: <transactiontype>,<document>, <anticipated transactiontype>,<anticipateddocument> 7.15
7.6.3 Message 3: tfacr200: <transactiontype>,<document>, <anticipated transactiontype>,<anticipateddocument> 7.15
7.6.4 Message 4: tfgld102: <fiscal year>, <batchnumber>, <transactiontype>, <document>, <correction transactiontype>,<correction document> 7.16
7.6.5 Message 5: tfgld106: <transactiontype>,<document>, <correction transactiontype>,<correctioninvoice> 7.16
7.6.6 Message 6: END OF CONVERSION 7.16
7.7 tccov2005m000: Changing enumerated field (tfcmg108) 7.16
7.7.1 Message 1: START OF CONVERSION tccov2005m000 by user <login name> 7.17
7.7.2 Message 2: tfcmg108: <payment batch>, <original company>, <payment type>, <supplier>,<order> 7.17
7.7.3 Message 3: END OF CONVERSION 7.17
7.8 tccov2006m000: Updating payment difference for anticipated document 7.17
7.8.1 Message 1: START OF CONVERSION tccov2006m000 by user <login name> 7.17
7.8.2 Message 2: tfacp200 <supplier>,<transactiontype>,<document>,<linenumber>, <transactiontype>,<document>,<linenumber>, <payment difference> 7.18
7.8.3 Message 3: tfacr200: <customer>,<transactiontype>,<document>, <linenumber>,<transactiontype>,<document>, <linenumber>, <payment difference> 7.18

7.8.4 Message 1: END OF CONVERSION *7.18*
7.9 tccov3001m000: Converting Manufacturing 3.1 to BAAN IV 7.18
7.9.1 Message 1: START OF CONVERSION tccov3001m000 by user <login name> 7.19
7.9.2 Message 2: END OF CONVERSION 7.19
7.10 tccov3002m000: Conversion Units to Units by Unit Sets 7.19
7.10.1 Message 1: START OF CONVERSION tccov3002m000 by user <login name> 7.19
7.10.2 Message 2: END OF CONVERSION 7.19
7.10.3 Message 3: tiitm001:<item.code>.<unit.set>.<units> 7.20
7.10.4 Message 4: tiitm002:<item.group>.<unit.set>.<units> 7.20
7.10.5 Message 5: tipcs021:<project.code>.<item.code>.<unit.set>.<units> 7.20
7.10.6 Message 6: tipcs002:<budget.code>.<calculation.part>.<unit.set> <units> 7.20
7.10.7 Message 7: tiedm010:<enginering.item>.<unit.set>.<units> 7.20
7.10.8 Message 8: tppdm025:<equipment.code>.<unit.set>.copied.from.itm001.<unit> 7.20
7.10.9 Message 9: tppdm025:<equipment.code>.Error:unit.not.in.unit.set. of.itm001 7.21
7.10.10 Message 10: tppdm025:<equipment.code>.<unit.set>.<units> 7.21
7.10.11 Message 11: tppdm035<subcontracting.code>.<unit.set>.copied.from.itm001.<unit> *7.21*
7.10.12 Message 12: tppdm035<subcontracting.code>.<unit.set>.copied.from.itm001.<unit> *7.21*
7.10.13Message 13: tppdm035:<subcontracting.code>.<unit.set>.<units> 7.21
7.10.14 Message 14: tppdm625:<project>.<equipment.code>.<unit.set>.copied.from.pcs021. <unit> *7.22*
7.10.15 Message 15: tppdm625:<project>.<equipment.code>.Error:unit.not.in.unit.set. of.pcs021 *7.22*
7.10.16 Message 16: tppdm625:<project>.<equipment.code>.<unit.set>.<units> 7.22
7.10.17 Message 17: tppdm635:<project>.<subcontracting.code>.<unit.set>.copied.from.pcs021. <unit> *7.22*
7.10.18 Message 18: tppdm635:<project>.<subcontracting.code>.Error:unit.not.in.unit.set. of.pcs021 *7.22*
7.10.19 Message 19: tppdm635:<project>.<subcontracting.code>.<unit.set>. <units> 7.23
7.10.20 Message 20: tppdm605:<project>.<project.item>.Error:item.not.in. pdm625 7.23
7.10.21 Message 21: tppdm605:<project>.<project.item>.Error:item.not.in. pdm635 7.23
7.10.22 Message 22: tppdm605:<project>.<itemcode>.<unit.set>.<units> 7.23
7.10.23 Message 23: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pcs021 *7.23*
7.10.24 Message 24: tppdm605:<project>.<itemcode>.<unit.set>.copied.from. pdm625 *7.24*
7.10.25 Message 25: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pdm625 *7.24*

7.10.26 Message 26: tppdm605:<project>.<itemcode>.<unit.set>.copied.from.pdm635 7.24
7.10.27 Message 27: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pdm635 *7.24*
7.10.28 Message 28: tppdm604:<project>.<itemcode>.<unitset>.copied.from.tppdm605 7.24
7.10.29 Message 29: tppdm604:<project>.<itemcode>.<unitset>.Error:unit.not.in.unit.set.of.pdm605 *7.25*
7.10.30 Message 30: Base Weight Unit not filled 7.25
7.10.31 Message 31: Base Length Unit not filled 7.25
7.10.32 Message 32: Base Area Unit not filled 7.25
7.10.33 Message 33: Base Volume Unit not filled 7.25
7.10.34 Message 34: Base Time Unit not filled 7.26
7.10.35 Message 35: parameters MCS not present 7.26
7.10.36 Message 36: physical quantity '<description>' has duplicates (conversion factor= 1.0) 7.26
7.10.37 Message 37: No units present 7.26
7.10.38 Message 38: base unit physical quantity '<description>' not known (conversion factor= 1.0) 7.26
7.10.39 Message 39: domain <domain.name>has no enumerated values 7.27
7.10.40 Message 40: Error <error.code> in qss.search(<search.mode>,<search.argument>,<table>,<definition>,<number>) *7.27*
7.10.41 Message 41: Error <error.code> in qss.sort(<table>,<definition>,<number>) *7.27*
7.11 tccov3003m000: Conversion Unit Conversion Factors 7.27
7.11.1 Message 1: START OF CONVERSION tccov3003m000 by user <login name> 7.27
7.11.2 Message 2: END OF CONVERSION 7.28
7.11.3 Message 3: CONFLICT: item ><item.code>><item.unit>><alternate.unit>><conversion.factor>> conflict general conversion 7.28
7.11.4 Message 4: CONFLICT proj/item ><project>><item.code>><unit>><purchase.unit>><conversion.factor>> conflict general conversion 7.28
7.11.5 Message 5: CONFLICT proj/item ><project>><item.code>><unit>><purchase.price.unit>><conversion.factor>>conflict general conversion 7.28
7.11.6 Message 6: CONFLICT proj/item ><project>><item.code>><unit>><storage.unit>><conversion.factor>> conflict general conversion 7.29
7.11.7 Message 7: tiitm004:<project>.<item.code>.<item.group>.<base.unit>.<unit>.<conversion.factor> *7.29*
7.11.8 Message 8: Base Weight Unit not filled *7.29*
7.11.9 Message 9: Base Length Unit not filled 7.29
7.11.10 Message 10: Base Area Unit not filled 7.29
7.11.11 Message 11: Base Volume Unit not filled 7.30
7.11.12 Message 12: Base Time Unit not filled 7.30
7.11.13 Message 13: parameters MCS not present 7.30

7.11.14 Message 14: physical quantity '<description>' has duplicates (conversion factor = 1.0) 7.30
7.11.15 Message 15: No units present 7.30
7.11.16 Message 16: base unit physical quantity '<description>' not known (conversion factor = 1.0) 7.31
7.11.17 Message 17: domain <domain.name>has no enumerated values 7.31
7.11.18 Message 18: Error <error.code> in qss.search(<search.mode>,<search.argument>,<table>,<definition>, <number>) *7.31*
7.11.19 Message 19: Error <error.code> in qss.sort(<table>,<definition>, <number>) *7*.31
7.11.20 Message 20: END OF CONVERSION; <number.conflicts> CONFLICTS FOUND *7.31*
7.11.21 Message 21: END OF CONVERSION; NO CONFLICTS *7*.31

About this document

This document explains how to convert TRITON 3 to BAAN IV. Two scenarios are possible: one in which the BAAN IV and TRITON 3 environment, exist next to each other for a period of time (two environments scenario), and one in which the TRITON 3 environment is updated to BAAN IV (single environment scenario).

Chapter 1 describes a number of general issues regarding the conversion from TRITON 3 to BAAN IV. These issues include the scope of the document, its target group, the two scenarios, and a number of remarks and recommendation, that you must take into account before running the actual conversion.

After reading this chapter and studying the flowcharts, you can decide which scenario to apply. You will also have an understanding of the tasks which have to be performed and their consequences.

Chapter 2 describes the steps needed to prepare TRITON 3 for the conversion to BAAN IV according to the two environments scenario.

Chapter 3 describes the steps needed to convert the prepared software components and data from TRITON 3 to BAAN IV according to the two environments scenario.

Chapter 4 describes the steps needed to update the TRITON 3 environment to BAAN IV and to perform the data conversion according to the single environment scenario.

Chapter 5 describes the last step of the conversion (for both the two and the single environment scenarios) and the sessions needed to perform a data check, after the conversion.

Appendix A describes the steps needed to upgrade both external and in-house customizations.

Appendix B supplies a list of (error) messages.

1 Introduction

This chapter describes a number of general issues regarding the conversion from TRITON 3 to BAAN IV.

1.1 Scope

This document explains how you can convert TRITON 3 to BAAN IV. It does not explain the concept of VRC structures and does not provide extensive help information on how to upgrade general or in-house customizations (C-VRCs and O-VRCs, respectively).

1.2 Target group

This document can be used both by Baan conversion teams and by Baan customers. For a flawless conversion from TRITON 3 to BAAN IV, extensive knowledge of both TRITON and BAAN is required.

1.3 Remarks and recommendations

Prior to the actual conversion, a number of considerations must be taken into account.

The conversion does not take multisite or multicompany conversion in consideration. Before going through a conversion scenario, check which tables are affected in your multicompany or multisite environment.

1.3.1 Conversion to BAAN IV: two possible scenarios

There are two options, single environment or two environments:

1 **Single**
The new version is installed by using one environment. In other words, all programs are updated with the new version and all data is converted to BAAN IV.

2 **Two**
An upgrade is done, using two different environments: an old one with the old version and a new one with a fresh installed version.

An upgrade or update is possible for:

Old version	New Version	
	BAAN IVa	BAAN IVb
TRITON 3.0	Two/Single	Two *
TRITON 3.1	Two/Single	Two/Single
BAAN IVa	x	Single

Note 1:

Before you can update a file from a TRITON 3.1 environment (single environment), you must define the UNIX variable **NO_PREINST** as follows:

NO_PREINST=1

export NO_PREINST

Note 2:

You can, temporarely, use the TRITON 3.*x* environment under BAAN IV Tools. Note that this time must be as short as possible.

Both scenarios have their advantages and disadvantages. The most important are the following:

Pros and cons of the two scenarios

Conversion Scenario	Advantages	Disadvantages
Single environment	◆ not too much extra diskspace required ◆ more simple conversion (few manual actions required) ◆ both the 'old' software version and the new software version can be used simultaneously in one environment (containing different data) ◆ less strain on the server (one server, one environ-ment)	◆ users have to leave the environment during update (software) ◆ some development actions are not possible anymore in the 3.0/3.1 package combinations after update
Two environments	◆ operational environment remains operational during installation/test conversion ◆ all actions are possible in the 3.0/3.1 (operational) environment during the whole procedure ◆ operational environment remains unchanged (reliable) during instal-lation/test conversion	◆ considerable more space is required (since two complete environments coexist for a period of time) ◆ the conversion has more steps, which are also more complex ◆ the conversion takes a considerable amount of time, and requires a lot of manual actions. ◆ in case of a one server - two environments situation, more strain is put on the server

The two conversion scenarios are referred to as the single environment scenario and the two environments scenario.

Both scenarios are outlined in a set of flowcharts, which are included in this document.

It is advisable to determine beforehand which of these two conversion methods is the best for the specific situation, and to study the steps of the corresponding flowchart(s). The charts are not entirely self-explanatory, so turn to the corresponding texts in this document for more detail.

It is important to understand the impact of all the choices, presented in the flowchart(s), especially those concerning customized table definitions (in case of a non standard data model (DM); see also section "Customizations").

The (data) correction program is suitable for converting data from the following TRITON versions to BAAN IV:

- 3.0b, 3.0b1, 3.1a, 3.1a1, 3.1b, and 31b2.

Note:

BAAN IVa is a restricted release, only the conversion to BAAN IVb has been tested on application level.

Apart from these conversion and correction programs, BAAN IVb requires you to run some small correction programs.

1.3.2 Customizations

To convert a standard TRITON 3 version to a standard BAAN IV version, you can use the standard conversion program. If the TRITON 3 version includes customized table definitions, and therefore the BAAN IV environment is assumed to these as well, the standard conversion program must be modified to fit the customer's specific configuration.

Note:

In the case of a nonstandard data model, the test conversion can only be carried out after the customizations have been redefined under BAAN IV, and the conversion program has been adjusted to the BAAN IV table definitions.

Check the derived-from structure for all possible types of non standard software. That is: all types of localizations (L-VRCs), line-of-business-specific software (B-VRCs), general customizations (C-VRCs) and in-house customizations (O-VRCs). If any of these are present, check whether these contain customized table definitions, and make sure the correction program is adjusted to these, if necessary.

Do not attempt a data conversion before this issue is resolved.

For more information on the tables which must be converted, see the online help information of the Conversion to BAAN IV (tccov0000m000) session.

1.3.3 Localizations

For the conversion of global localizations, version 3.1b is required, which means that TRITON 3.1a users must first be converted to 3.1b. For the conversion of other localizations, see the previous section.

1.3.4 Disk space

Two environments scenario

After BAAN IV has been installed, two different environments exist: the operational TRITON 3 environment and the new BAAN IV environment. The data in the old environment must be copied to the new environment.

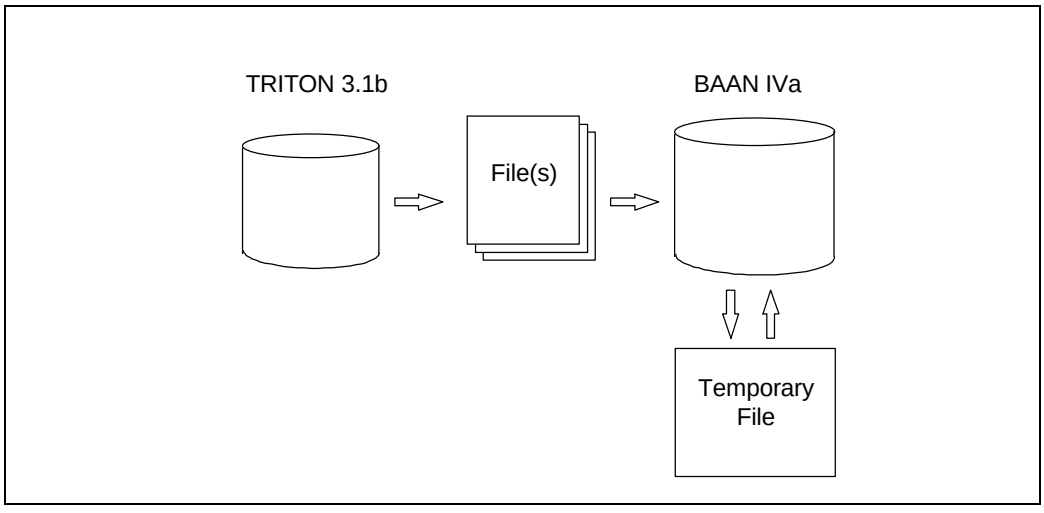

Figure 1-1 Conversion to BAAN IV

During the conversion to BAAN IV, temporary files are used to store the conversion data of each separate table.

1.3.5 Straight migration

In case of a straight migration, the business functions which are used in TRITON 3 are migrated to BAAN IV, where they will be used in the same way (if possible). Consequently, a straight migration offers maximum security, while the complexity of the project is reduced because new functionality is not implemented until later.

1.3.6 Test conversion

The application tables (data) in TRITON 3 are converted to BAAN IV by one of the conversion specialists of Baan or the customer’s own conversion specialists.

The test conversion simulates the final conversion, with the following objectives:

- Attune the conversion environment to customizations (if any).
  In case the TRITON 3 version includes customized table definitions, the standard conversion program must be modified to fit the customer's configuration. In this case, the test conversion can only be carried out after the customizations have been redefined under BAAN IV.
- Provide information about the run time of the actual conversion.
  The planning of the final conversion is based on the run time of the test conversion. After the final conversion has been started, no new transactions must be created in TRITON 3, because these transactions will not be included in the conversion.
- Provide certainty about the progress of the conversion run.
  The final conversion must take place without errors. If any error occurs during the test conversion, its causes must be traced and cleared so that the test conversion can be repeated without having the error block the final conversion.
- Provide certainty about the results of the conversion run.
  The results of the test conversion are assessed based on the data and procedure tests. Because all functions and procedures in the new BAAN IV application can be used, based on the data of the users own company, maximum certainty is achieved regarding the continuity of business processes.

1.3.7 Integration of implementation and data conversion

Data conversion must be an integral part of the installation and is, therefore, included in the plan of action.

This plan of action includes the following aspects:

- Upgrade or installation of hardware
- Installation of BAAN IV (if applicable)
- Training of end users (BAAN IV tools and applications)
- Development of customizations
- Implementation of BAAN IV Tools
- Preparation of test conversion
- Test conversion
- Data test
- Procedure test
- Go/No Go decisions
- Final conversion

1.4 Terms and abbreviations

Term/Abbreviation	Description
BSE/$BSE	BSE refers to the BAAN Shell Environment. By means of the BSE variable, a short link is established which refers to the directory in which all BAAN programs and directories are located.
Environment	The technical area which consists of the directories on a system used by the programs.
Data Conversion	The conversion of all company data to a new format.
Software Conversion	The conversion and upgrade of general or in-house customized software components to a new version.
VRC	The Version Release Control (VRC) module in BAAN Tools is used to manage different versions and releases of a package. There are 4 types of VRCs: L: localizations B: line-of-business-specific software C: general customizations O: in-house customizations

1.5 References

For more information on the tools and utilities that can be used in the conversion procedure of customized software, see the following documents:

- New Functions in BAAN IV 5972 US
- New Functions in BAAN IVb P3000C US
- BAAN Utilities User Manual 5569 US
- Component Merge Tool (CMT) module Module Description 7795 US
- Installation Guide for BAAN IV on Unix U7016A US
- ORACLE Installation Guide for BAAN IV U7028B US
- INFORMIX Installation Guide for BAAN IV U7029A US
- DB2/6000 Installation Guide for BAAN IV U7030A US
- BAAN Tools User Manual 5564 US

2 Preparing TRITON 3 for conversion: the two environments scenario

This chapter describes the steps needed to prepare TRITON 3 for the conversion to BAAN IV.

2.1 Steps in the procedure

The following steps can be distinguished to prepare software components for conversion:

1 General
2 Software export
3 Data export

Each of these main steps may consist of a number of additional steps.

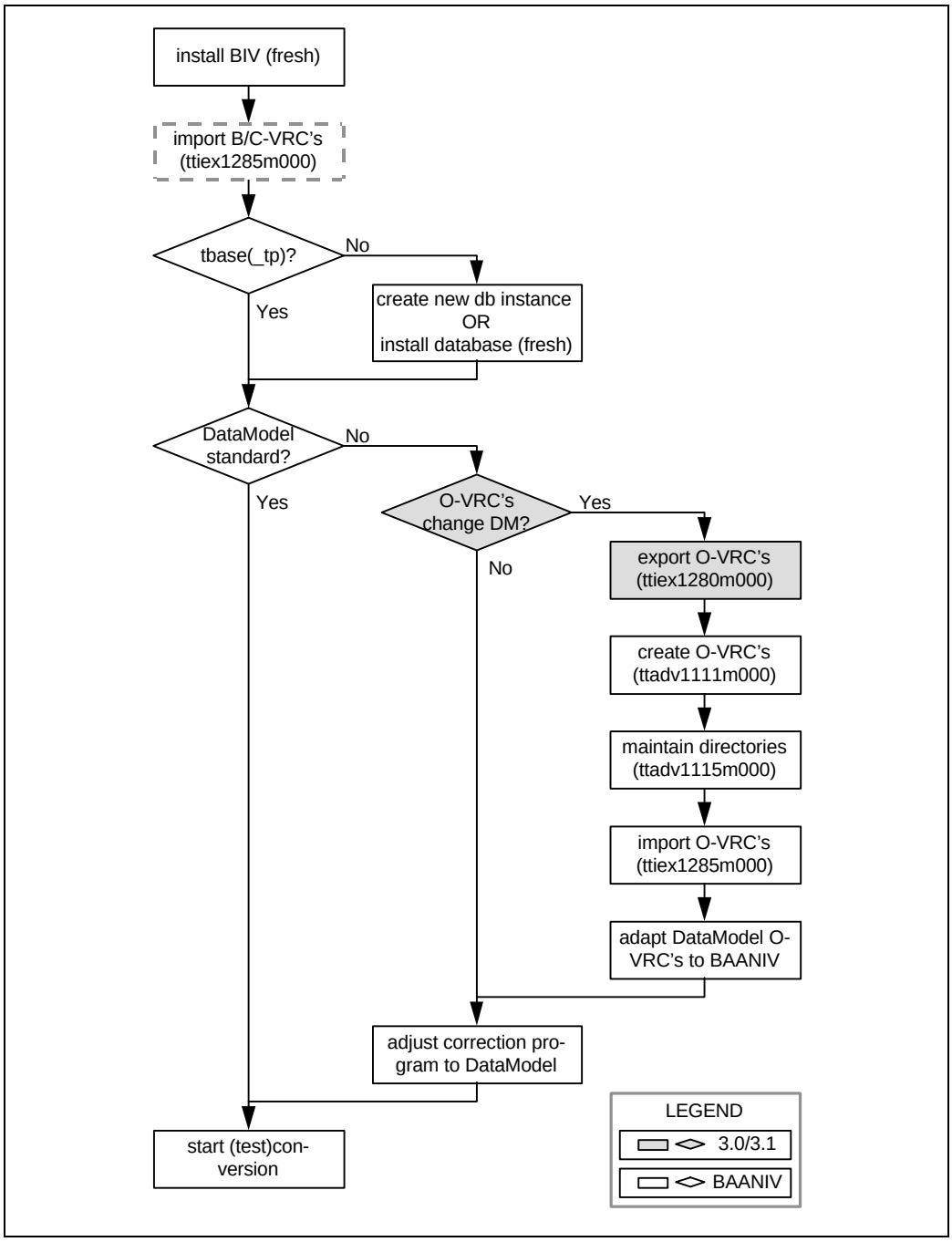

Figure 2-1 Preparing TRITON 3 for conversion: the two environments scenario

2.2 General

2.2.1 Checking the consistency of the TRITON 3 data

If the item data in Triton 3.0b/3.0b1 are inconsistent, the conversion to BAAN IV will fail. For this reason you must run a query before you start the actual conversion. This query will detect inconsistencies in the TRITON database and is described in the procedure below.

Note:

You must carry out the query for each company to be converted.

1 Use **vi** to create a file, for example **check_db**, containing the query below.

```
select tdpur036.suno, tdpur036.item
from tdpur036
where not exists
(    select tiitm001.item
     from    tiitm001
     where tiitm001.item = tdpur036.item          )
```

2 Define and export the following environment variables from the UNIX command line:

- **BSE**
- **BSE_TMP**
- **PATH**

3 Check that the user, carrying out the query, is linked to the same package combination to which the company to be checked is linked.

4 Execute the query by entering the following command.

qptool6.1 -f check_db -C100

5 If the query generates output, there are items in **tdpur036**, that are not present in the item file **tiitm001**. You must either remove these items from **tdpur036** or add them in **tiitm001** by using session tiitm0101m000.

6 Repeat the steps 1 to 4 for the query printed below. Note that you must take another name for the second query.

```
select tdsls037.cpls, tdsls037.item
from tdsls037
where not exists
(    select tiitm001.item
     from    tiitm001
     where tiitm001.item = tdsls037.item)
```

7 If the query generates output, there are items in **tdsls037**, that are not present in the item file **tiitm001**. You must either remove these items from **tdsls037** or add them in **tiitm001** by using session tiitm0101m000.

2.2.2 Create a backup of the TRITON software and data

A backup must be created of the operational TRITON software and data.

2.2.3 Export Users/Authorizations/Devices/Currencies/Logical Tables (ottcon61export)

At this point, the data regarding the current users, authorizations, currencies, and devices is exported by means of the following steps:

Copy the ocon61export program from the BAAN IV environment
The ocon61export program must be copied from the new BAAN IV **$BSE/tools/ttB40_a/ottcon** environment to the TRITON environment. For TRITON 3.1 to the **$BSE/tools/tt6.1_*/ottcon** directory, and for TRITON 3.0 to the **$BSE/tools/tt6.0_*/ottcon** directory. This program is referred to as an object and does not have to be patched.

For example:

```
cp /baan4/bse/tools/ttB40_a/ottcon/ocon61export• /triton31/bse/tools/tt6.1_a/ottcon/ocon61export↵
```

The permissions and the owner of the ottcon61export program must be set to **bsp**.

Execute the ottcon61export session
The ottcon6.1export session in TRITON 3 is used to export all users, authorizations, currencies, devices, and logical databases to an ASCII file. This program can be started from the command line of a menu by typing **p** followed by **ottcon61export**.

```
Export Users Data              y/n:  y

Export Authorizations          y/n:  y

Export Currencies              y/n:  y

Export Device Data             y/n:  y

Export Logical Tables          y/n:  y




Output File                       :/usr8/bse/tmp/baaniv


Continue                       y/n:  y
```

2.3 Software export

This step only applies to customized software components.

2.3.1 Export Data Dictionary (ttiex1280m000)

This session is used to export the customized VRCs from TRITON 3 to an ASCII file. For each VRC, a separate export directory must be created.

```
ttiex1280m000                    alphanum     single occ (4)     Form 1-4 >
Export Data Dictionary




Package           : tc               TRITON Common
VRC               : 3.10   a  cv02   Challenge 3.0


Update            : Yes
Compared with VRC : 3.11  a


Export Based on Sessions          : No
Only components of package itself : Yes
Session  (from/to)             :              - zzzzzzzzzz
Languages                         : 1  2  3
Deliver Session from Status       : Developing
Create Components List             : Yes
Errors to Printer                 : Yes
Path Dumps        : /usr1/triton6/bse/export/tc


                                                          Choice: ..
```

2.4 Data export

There are two ways in which the company data can be exported or transferred to the new BAAN IV environment, namely by means of a general data export or by means of a database-specific data transfer.

2.4.1 General data export

The Create Sequential Dump of Table (ttaad4226m000) session can be used independently, of any database to create a sequential dump of the database tables.

```
ttaad4226m000                                  single occ (4)      Form 1-1
  Create Sequential Dump of Table
                                  From       To
  Company                        : 100     - 100
  Package                        :         - zz
  Table                          :         - zzz999
  Suppress Error Messages        : No
  Recover Isam File              : No
  Drop Table after Dump          : No
  Create Backup before Dropping  : No
  Specify Database Driver        : No
  Create Fixed Length Records    : No
  Specify Field Separators       : No
  Directory for Table Files      : No
  Basic Name for Dump File(s)    : /tmp/dump
  Name of the Error File         : 100.error

  Name of the dumpfile(s) will be:
  /tmp/dump.<company>                                          Choice: .
```

The advantage of this approach is that the company data remain operational. The disadvantages are the amount of disk space which is needed for the export file and the new database as well as the amount of time needed to export and import the company data.

2.4.2 Database-specific data transfer

A database-specific data transfer is a faster conversion method than the general export option. There are different types of database-specific data transfers, depending on the database type.

TBASE or TBASE_TP
For this method the database tables for TBASE and TBASE_TP must be copied to the BAAN IV environment and registered in BAAN IV (this second part is described in the import section of the next chapter). For TRITON 3.0 an additional step is necessary in the import (conversion of the tables).

Make sure that nobody is using the companies when you are copying them.

The advantage of this approach is that there is a faster conversion which mainly results from the fact that two procedure steps are omitted, namely the Create Sequential Dump of Table (ttaad4226m000) session and the Create Table from Sequential Dump (ttaad4227m000) session (see Chapter 3, Converting to BAAN IV: two environments scenario).

Informix or Oracle
The approach for these RDBMS highly depends on the customer's specific situation and is, therefore, too complex to be discussed in the context of this document. For more information, contact the local BAAN supplier, who can in turn contact the ISC for assistance. Partners and dealers in the EMEA area can contact the department of Technical Services of BAAN Netherlands.

3 Converting to BAAN IV: two environments scenario

This chapter describes the steps needed to convert the prepared software components and data from TRITON 3 to BAAN IV.

3.1 Steps in the procedure

The following steps can be distinguished in converting the TRITON 3 data and software to BAAN IV:

1 General
2 Software conversion
3 Data conversion

Each of these main steps may consist of a number of additional steps.

Next some flowcharts display the conversion in a schematic form.

3 Converting to BAAN IV: two environments scenario

This chapter describes the steps needed to convert the prepared software components and data from TRITON 3 to BAAN IV.

3.1 Steps in the procedure

The following steps can be distinguished in converting the TRITON 3 data and software to BAAN IV:

1 General
2 Software conversion
3 Data conversion

Each of these main steps may consist of a number of additional steps.

Next some flowcharts display the conversion in a schematic form.

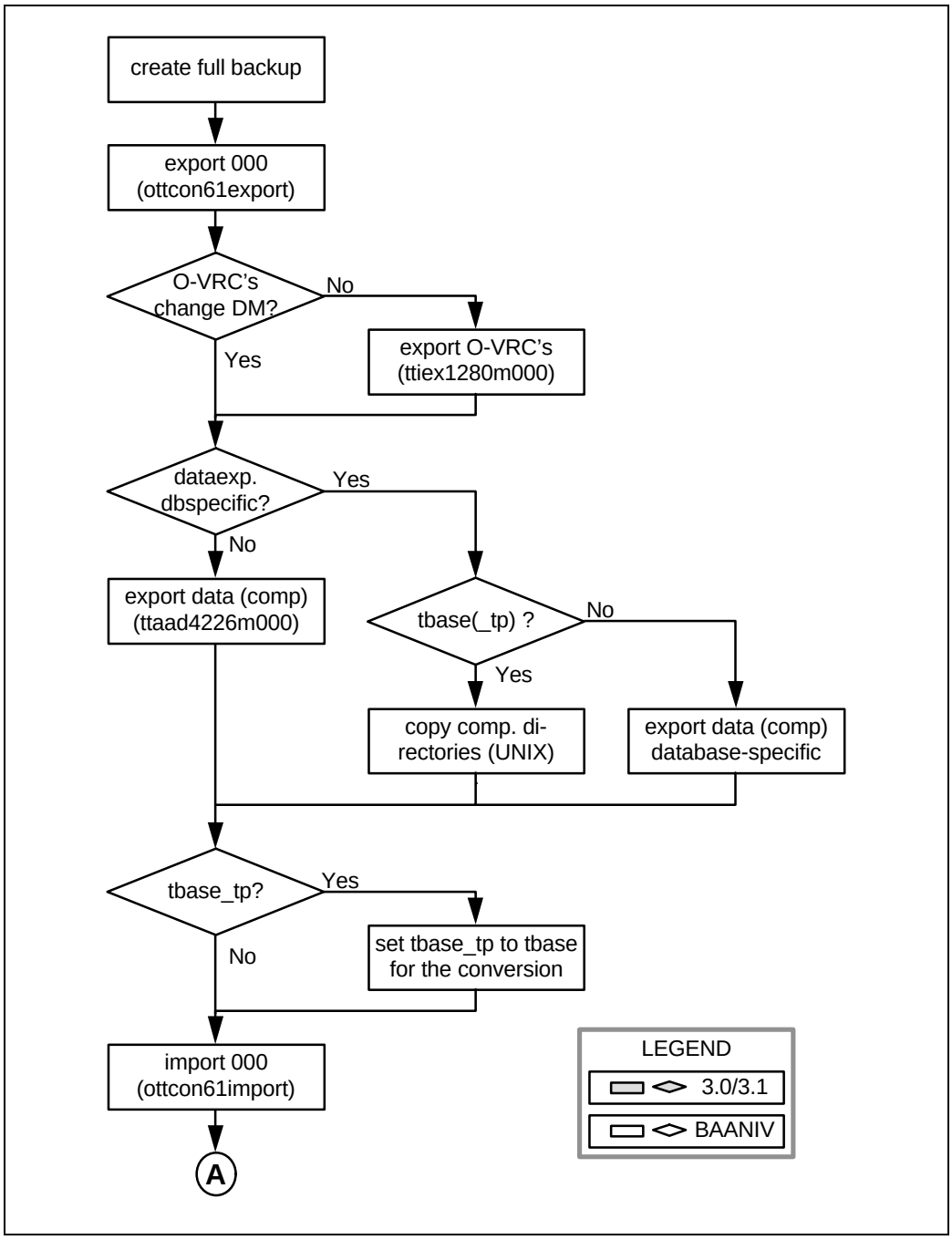

Figure 3-1 of 4 Converting to BAAN IV: two environments scenario

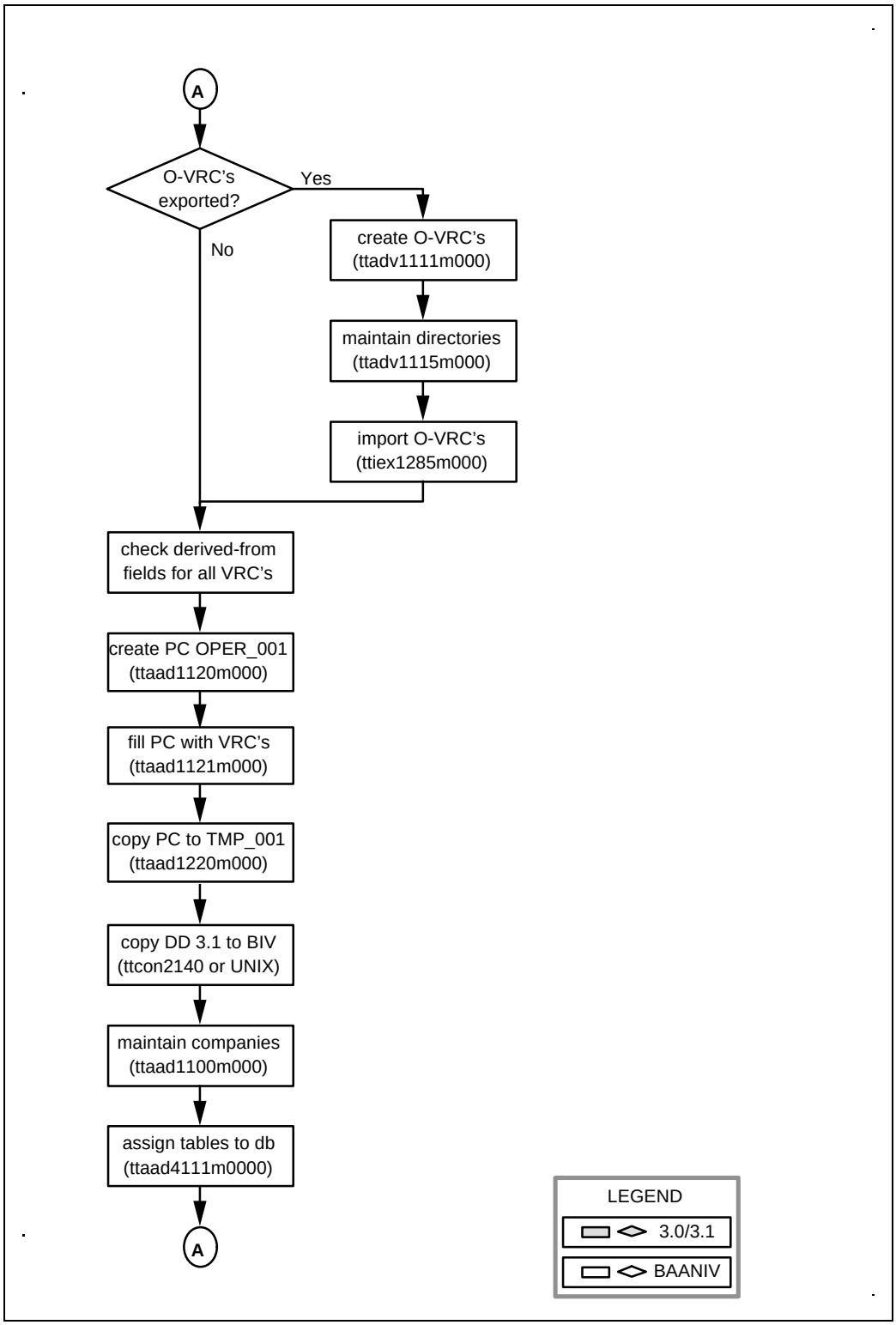

Figure 3-2 of 4 Converting to BAAN IV: two environments scenario

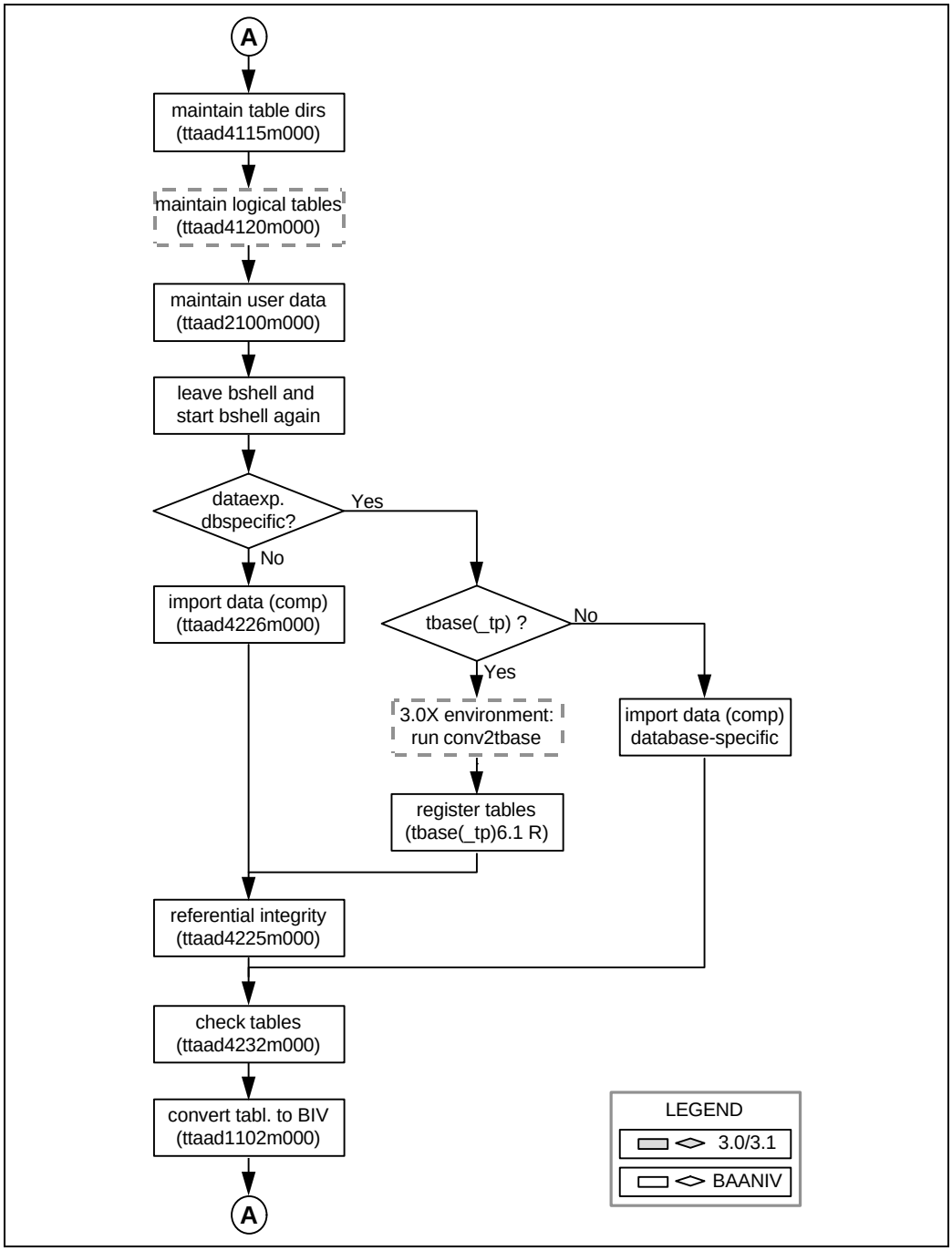

Figure 3-3 of 4 Converting to BAAN IV: two environments scenario

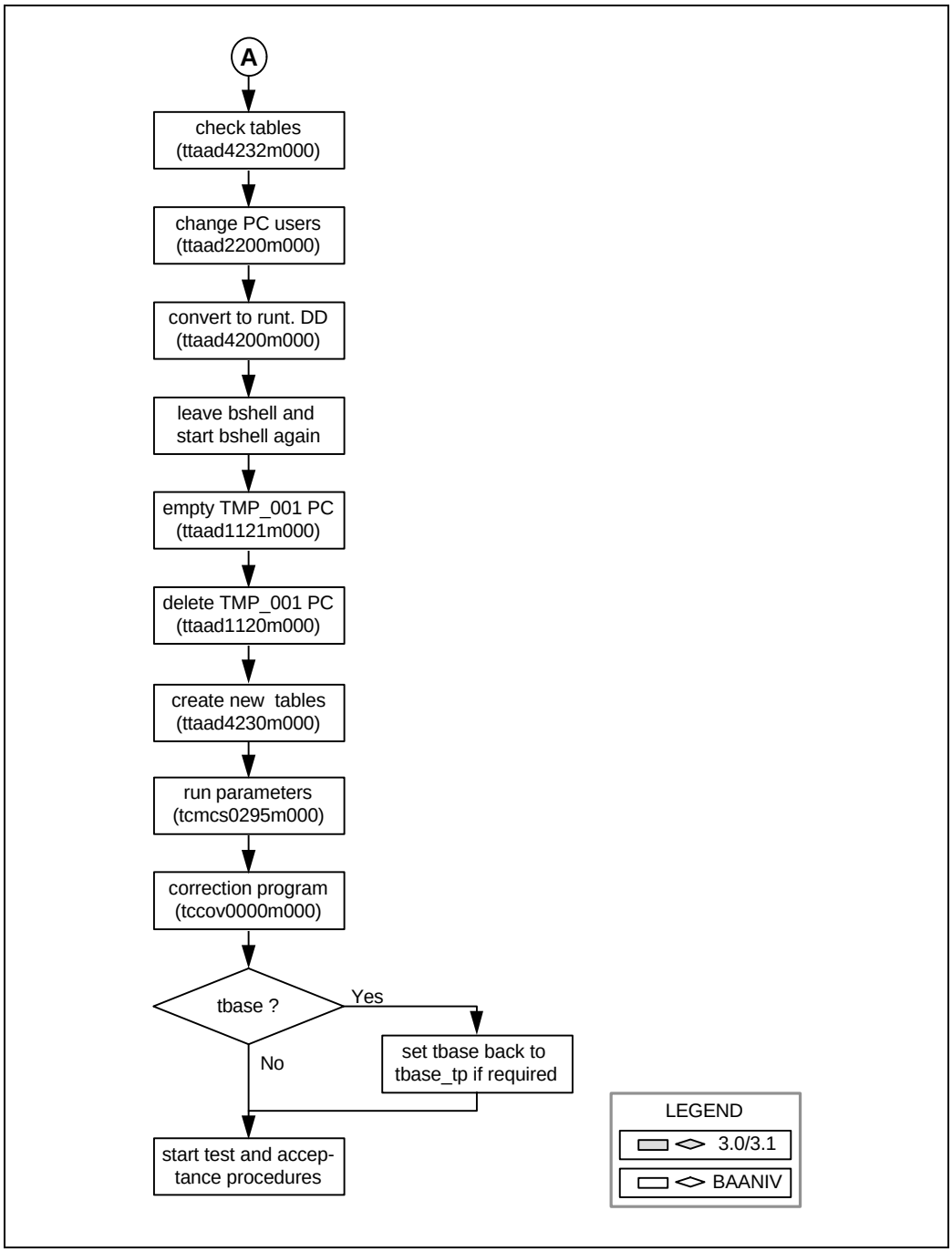

Figure 3-4 of 4 Converting to BAAN IV: two environments scenario

3.2 General

This step is executed from within the data conversion menu of BAAN IV:

```
mttcon2130m000                          Menu 1-1

      B A A N IV T o o l s              Date: 20-05-96

                              Company: 000

      Data Conversion 6.1/3.1 -> BAAN IV



     1 Copy Package Combination
     2 Copy datadef files from 3.1 to Temporary Package Combination
     3 Maintain Companies
     4 Assign Tables to Databases
     5 Maintain Table Directories (BAAN Base (TP))
     6 Maintain Logical Tables
     7 Maintain User Data

     8 Create Table from Sequential Dump
     9 Check Tables
    10 Convert 3.1 b tables to BAAN IV
    11 Import Users/Authorizations/Devices etc. from 6.1
                              Choice: 11
```

3.2.1 Import Users/Authorizations/Devices/Currencies/Logical Tables (ttcon61import)

At this point, the file which was saved in TRITON Tools 6.1 is imported. Instead of importing the old definitions, it is also possible to define new users, authorizations, devices, currencies, and logical tables.

```
ttcon61impor      alphanum   single-occ (4)   Form 1-1

 Import Users/Authorizations/Devices etc. from 6.1



 Path 6.1 Data       :
      /usr8/bse/tmp/baaniv




                                 Choice:
```

3.3 Software conversion

3.3.1 Import Data Dictionary (ttiex1285m000)

This session is mandatory for customized software components and is used to import the customized VRCs which originate from TRITON 3 in the proper B40 VRCs.

```
ttiex1285m000                 single-occ (4)   Form 1-1


 Import Data Dictionary  : /usr1/TRITON6/bse/export/tc



 Path Sequential Dumps  : /tmp

 Path Runtime Dictionary : ${BSE}/application

 Overwrite        : Yes

 Delete dumps
 after installation    : No



 Write software components: Yes
 in other Package VRC
 Package VRC       : tc TRITON Common    B400 a cv01




                                Choice:.
```

Example:

tc3.1O_a_cu02 --> tcB40O_a_cu02

3.3.2 Maintain Packages by Package Combination (ttaad1121m000)

This session is mandatory for customized software components and is used to create an operational package combination in BAAN IV which is referred to as OPER_001. Prior to converting the data tables, the data model must be assessed as being the correct data model. In addition, the derived structure of the VRCs must be determined.

For more information on how to upgrade customized VRCs, see Chapter 6, Upgrading Customizations.

3.4 Data conversion

3.4.1 Copy Package Combination (ttaad1220m000)

This session is used to copy the standard or customized package combination of BAAN IV (referred to as OPER_001) to a new temporary package combination (referred to as TMP_001). These package combinations must have the same VRC structure and the same data model (data definitions).

3.4.2 Copy data definition files

The data definition files are located in the directory which was specified by means of the Maintain Package Combinations (ttaad1120m000) session in the TRITON 3 environment. The target directory of the TMP_001 package combination can be determined by means of the same session in the BAAN IV environment.

If the TRITON 3 and BAAN IV environments are located on the same system, the Copy datadef files from 3.1 to Temporary Package Combination (ttcon2140m000) session must be used to copy the data definition files of the OPER_001 operational package from TRITON 3 to the TMP_001 temporary package combination in BAAN IV.

```
ttcon2140m000                 single-occ (4)   Form 1-1

Copy datadef files from 3.1 to Temporary Package Combination




 BSE-environment 3.1      : /usr1/TRITON6/bse

 Package Comb. 3.1 environment : OPER_001



 Temporary Package Combination : TMP_001

                                Choice:.
```

If the TRITON 3 and BAAN IV environments are located on different systems, there are two options to copy the data definition files.

The first option is to manually transfer the data definitions to the new system. This means that the existing data definitions in the temporary TMP_001 BAAN IV package combination are removed, after which the TRITON 3 data definition files are transferred or copied to the BAAN IV system.

In the example below, a tape is used to transfer this data to the BAAN IV system:

```
System I   TRITON 3 installed in                    /TRITON/bse
        Dictionary installed in                     /TRITON/dict
        Operational Package Combination                     OPER_001
System II  BAAN IV installed in                     /baaniv/bse
        Dictionary installed in                     /baaniv/dict
        Operational Package Combination                     TMP_001
Step 1
Copy the data dictionary files to tape:
Login as root on system I
# cd /TRITON/dict/ddOPER_001
# find * -print | cpio -ocvudB > /dev/<device>       # device of your own system
                    # for AIX rmt0 or rmt1
                    # for HP rmt/0m
Step 2
Login as root on system II
Put the tape from system I into the tapestreamer of system II
# cd /baaniv/dict/ddTMP_001
# rm -r d*                  # remove all existing data
                    # in this directory
# cpio -icvudB </dev/<device>     # device of your own system
                    # for AIX rmt0 or rmt1
                    # for HP rmt/0m
# chown -R bsp:bsp d*             # setting permissions
```

The other option is to mount the system with the TRITON 3 environment on the BAAN IV system, after which the procedure for two environments on the same system.

3.4.3 Maintain Companies (ttaad1100m000)

This session is used to define the companies which must be converted, without creating the tables. These companies are linked to the temporary package combination by means of the Package Combination field.

3.4.4 Assign Tables to Databases (ttaad4111m000)

This session is used to check the data definitions. Any changes which were made in this session must be converted to run time prior to exiting the session.

3.4.5 Maintain Table Directories (BAAN Base (TP)) (ttaad4115m000)

This session is used to check the table directories. These table directories must refer to the new path for the BAAN IV tables that must be converted from TRITON 3. Any changes that were made in this session must be converted to run time prior to exiting the session.

This is an optional session which is used for the BAAN Base database program (tbase or tbase_tp). Consequently, this session must not be used for Oracle or Informix databases.

3.4.6 Maintain Logical Tables (ttaad4120m000)

This session is used to define the logical companies. This data can also be exported and imported by means of the Export Users/Authorizations/Devices/Currencies/Logical Tables (ottcon61export) session and the Import Users/Authorizations/Devices/Currencies/Logical Tables (ttcon61import) session.

3.4.7 Maintain User Data (ttaad2100m000)

This session is used to link the logon name of the user who executes the conversion to the TMP_001 temporary package combination.

Any changes that were made in this session must be converted to run time prior to exiting the session. In addition, the bshell must be restarted.

3.4.8 Data import

There are two ways in which the company data can be imported in or transferred to the new BAAN IV environment, namely by means of a general data import or by means of a database-specific data transfer.

General data import

The Create Table from Sequential Dump (ttaad4227m000) session can be used independently of any database to import the TRITON 3 tables from the old environment into the BAAN IV environment. This import must be executed in the TMP_001 temporary package combination.

For this type of import, set the All Parameters field to No, and set the Ignore Referential Integrity Constraints field to Yes.

On the second form of this session, the file name for the error log file must be specified.

Database-specific data transfer

A database-specific data transfer is a faster conversion method than the general export option. There are different types of database-specific data transfers, depending on the database type.

TBASE or TBASE_TP

The database tables for TBASE and TBASE_TP must be copied to the BAAN IV environment and registered in BAAN IV.

For TRITON 3.0 and TRITON 3.1 different steps are necessary.

TRITON 3.1
The database tables for TBASE and TBASE_TP must be registered in BAAN IV. This is done by means of the **register6.1** script, which must be run under the tables directory. This script must be located in the **$BSE/bin** directory while its permissions must be set by means of the **chmod 755 $BSE/bin/register6.1** command.

```
for i in ` find * -name *.bdt -print `
do
    tbase6.1 R r ${I}        # Use tbase_tp6.1 for TBASE_TP database !
done
```

TRITON 3.0
For TRITON 3.0 an additional step is necessary. Make sure all TRITON 3.0 tables are registered correctly at the beginning of this step.
Set **$BSE** to the TRITON 3.0 environment and make a list of all the existing tables, but not the dictionary tables:

```
$ tbase6.0 R d | awk '{print $1".bid"}'|grep -v "dict" > /usr/tmp/conv_list
$ tbase6.0 R d | awk '{print $1".bid"}'|grep -v "dict" > /usr/tmp/regr_list
```

Set **$BSE** to the BAAN IV environment now, and convert all tbase tables into the new tbase format (**tbaseconv6.1** is located in **$BSE/bin** of the BAAN IV environment):

```
$ tbaseconv6.1 bsp /usr/tmp/conv_list
```

Now register all tbase tables for BAAN IV. This is done by means of the **register6.1** script which must be run under the tables directory. This script must be located in the **$BSE/bin** directory while its permissions must be set by means of the **chmod 755 $BSE/bin/register6.1** command.

```
for i in ` find * -name *.bdt -print `
do
    tbase6.1 R r ${I}        # Use tbase_tp6.1 for TBASE_TP database !
done
```

Informix or Oracle
The approach for these RDBMS highly depends on the customer's situation and is, therefore, too complex to be discussed in the context of this document. For more information, contact the local BAAN supplier who may contact the ISC for assistance. Partners and dealers in the EMEA area can contact the department of Technical Services of BAAN Netherlands.

3.4.9 Check Tables (ttaad4232m000)

This session is used to check the tables that must be converted for consistency in relation to table definitions. If there are no inconsistencies, no error messages will occur other than the Table Does Not Exist message related to error 506, which is prompted by the new tables in BAAN IV. These tables can be created after the conversion.

Since the tables are locked after this session has been used, the bshell must be restarted after this session is closed.

3.4.10 Convert 3.0/3.1 Tables to BAAN IV (ttaad1102m000)

This session is used to convert the table format from the temporary package combination to the operational package combination.

3.4.11 Check Tables (ttaad4232m000)

After the tables have been converted, this session is used to check the tables for consistency in relation to table definitions. If there are no inconsistencies, no error messages will occur. Since the tables are locked after this session has been used, the bshell must be restarted after this session is closed.

3.4.12 Change Package Combination for Users (ttaad2200m000)

This session is used to link the users to the new, operational BAAN IV package combination which is referred to as OPER_001.

3.4.13 Convert User Data to Runtime Data Dictionary (ttaad4200m000)

This session is used to convert the modified user data to the run time data dictionary.

After this session has been used, the bshell must be restarted. Use a user name which is linked to the OPER_001 operational package combination.

3.4.14 Maintain Package Combinations (ttaad1120m000)

This session is used to delete the TMP_001 temporary package combination. This is done by means of the **Delete** option on the **Edit** menu.

3.4.15 Create Tables (ttaad4230m000)

This session is used to create the new BAAN IV tables for specific companies and tables.

3.4.16 Reorganize Tables (ttaad4225m000)

This session is used to repair the reference counters of the imported tables. Do this for all involved companies by entering the following fields:

Data and Indices	No
Reference Integrity field to	Yes
Repair Reference Counters field to	Yes
other fields	No

In addition, the errors in the $BSE/log/log.aad4225 logfile must be checked. Since the approximately 200 tables that were added to BAAN IV are not yet available, the No DD Found For... message will appear several times.

4 Converting to BAAN IV: the single environment scenario

This chapter describes the steps needed to convert the prepared software components and data from TRITON 3 to BAAN IV.

4.1 Steps in the procedure

The following steps can be distinguished in converting the TRITON 3 data and software to BAAN IV:

1 Preparation
2 Installation procedure

Each of these main steps may consist of a number of additional steps.

The following flowcharts outline the conversion.

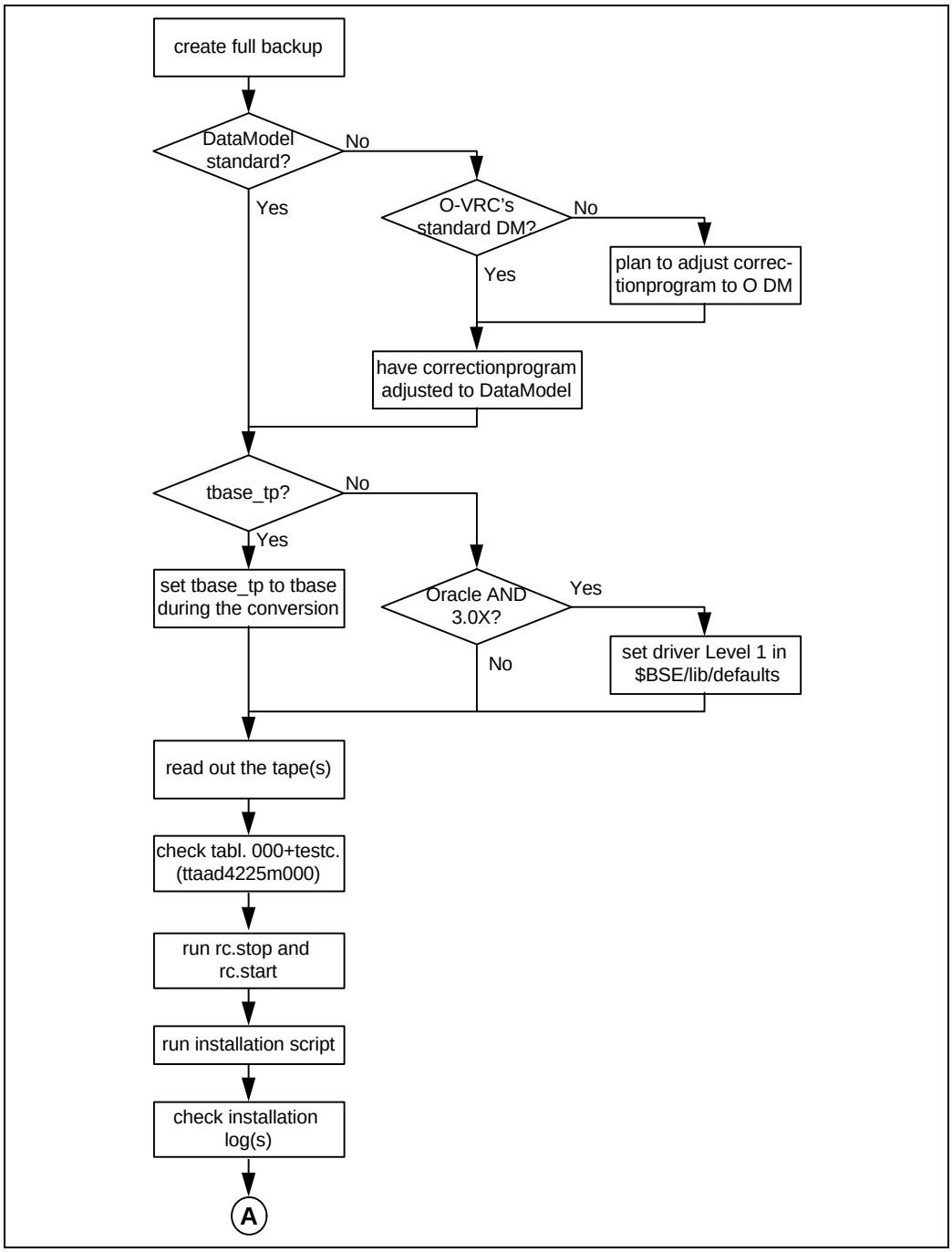

Figure 4-1 of 3 Converting to BAAN IV: the single environment scenario

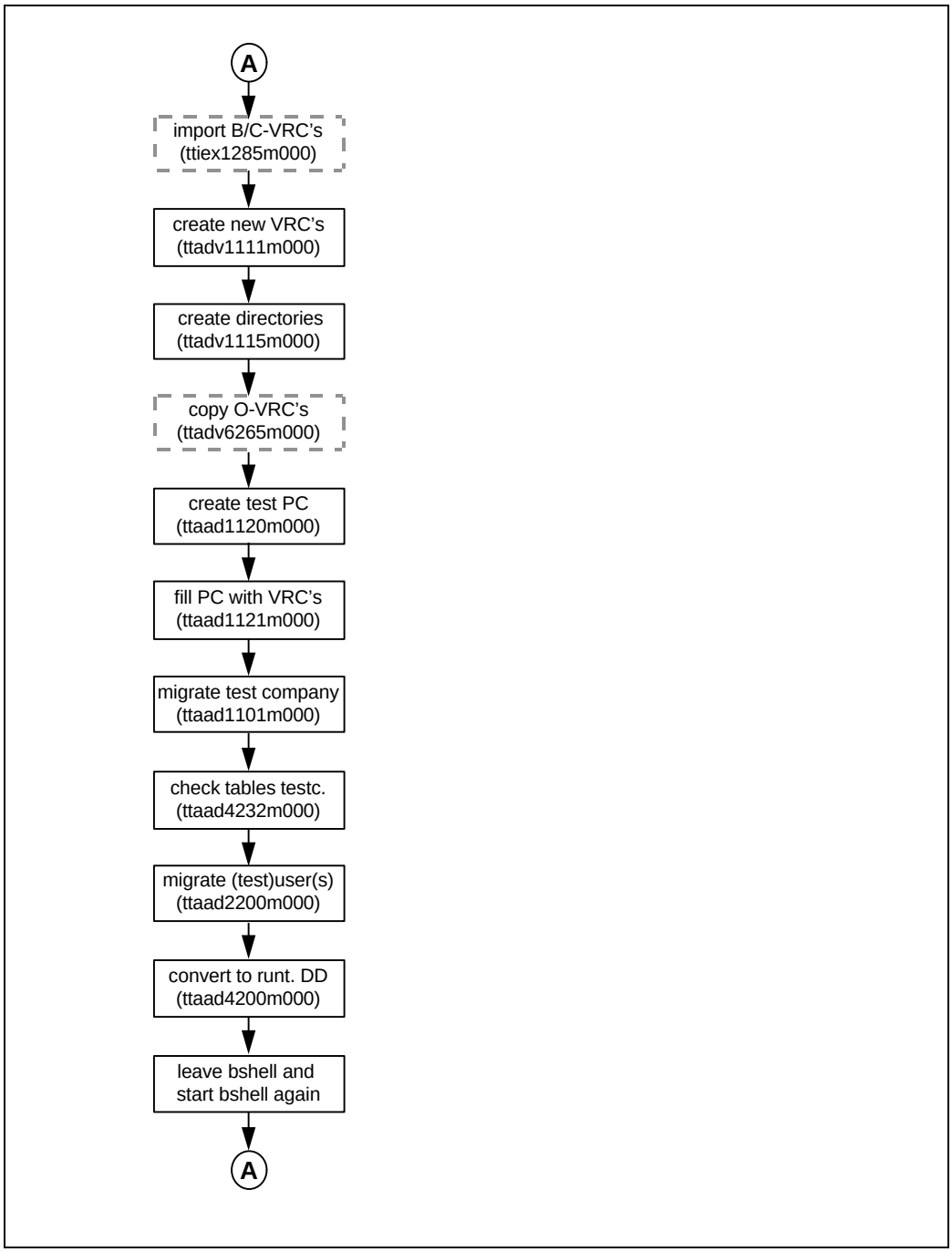

Figure 4-2 of 3 Converting to BAAN IV: the single environment scenario

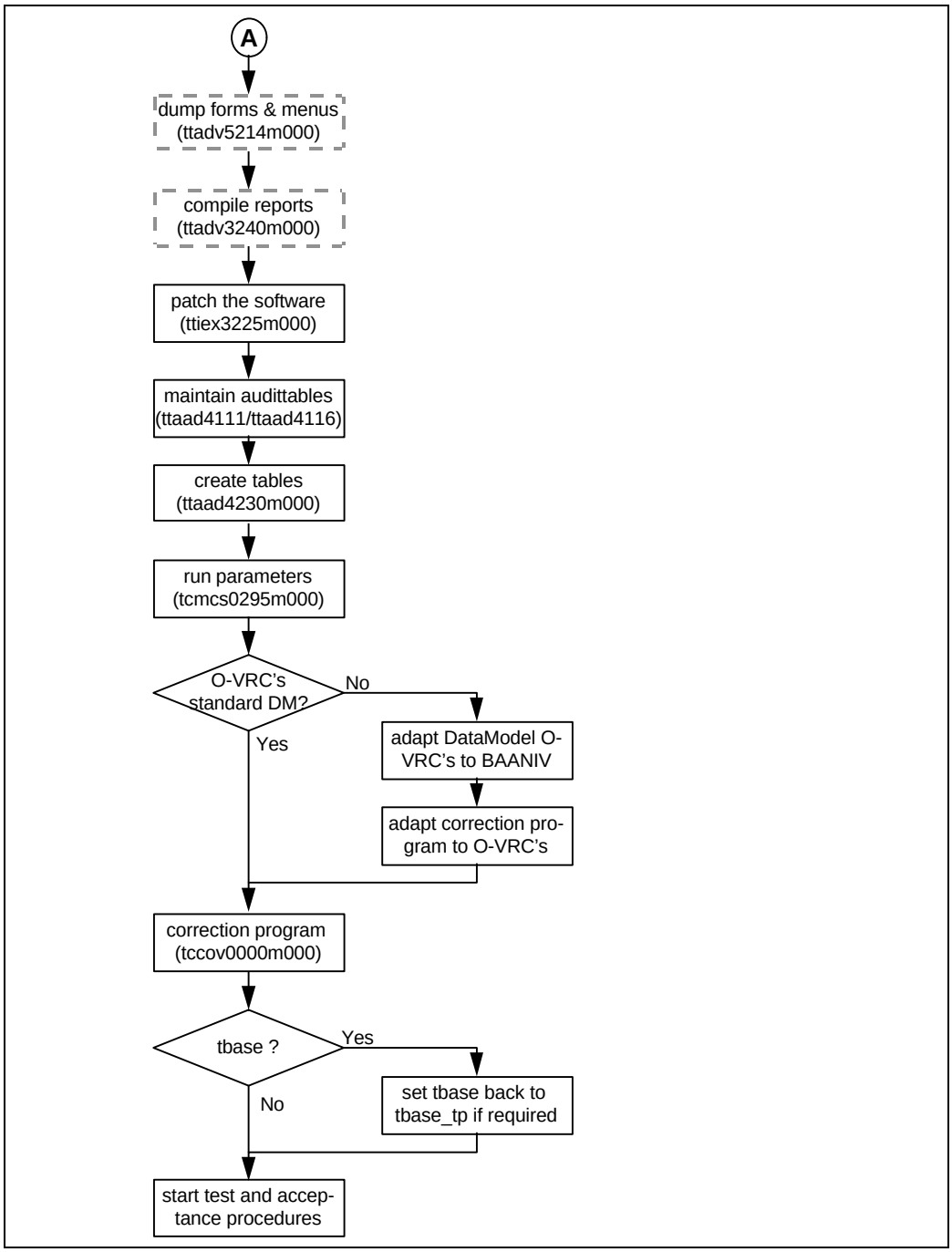

Figure 4-3 of 3 Converting to BAAN IV: the single environment scenario

4.2 Preparation

4.2.1 Backup

The customer must have a good, recent backup of the software and the tables.

4.2.2 TRITON

While the installation script is running, the customer cannot use TRITON. Furthermore, you must run correction programs before companies migrated to BAAN IV can be used for testing.

4.2.3 Time frame

The time frame of the update very much depends on the system and database types. Assuming that the installation script is started on the evening of the first day, allow 4 to 8 hours on the second day to complete the job. This may be considerably more—to the point of having to allow a third day—if Informix or Oracle installations and/or the conversion of many or large test companies are involved.

The time frame on the first day is 1 to 5 hours to read out the tape(s); a master tape takes longer than a normal installation tape. Allow an additional half hour to prepare the update and to discuss the procedure with the customer. Try to take into account that the customer may want to run a backup at the last moment. The installation script takes between 3 hours and several days (in case of logged databases on a relatively small system) to run.

4.2.4 Prerequisites

Reserve 300 MB to 400 MB for reading the tape. A full update requires 350 to over 450 MB (for one language, excluding customization, under bisam). Allow an extra 200 MB per additional language.

Determine if there is sufficient disk space on the system at the proper places (see previous section).

Check the derived-from structure, the package combinations, and associated companies.

After installation, the existing TRITON 3 structure (that can still be used) must be copied into BAAN IV. This requires the creation of at least one new BAAN IV package combination for testing.

4.2.5 Test

This test package combination is referred to in this document as TMP_001, and is a copy of the operational package combination in TRITON 3 known as OPER_001.

The environment must be set up for testing; this implies that you should not at this stage migrate all existing package combinations (and associated company numbers) to BAAN IV! When the test conversion is satisfactory, later on only those BAAN IV equivalents of the package combinations have to be created that have companies attached to them (but do not create an equivalent of the standard [31Sa or 31Sb] package combination).

4.2.6 Customizations

Make sure that for the conversion of own software, all that software is included in the process. Take other customizations into account as well (C-VRCs and B-VRCs). If there is customized software in TRITON 3, the customization supplier must migrate this to BAAN IV. The upgrade cannot take place before that migrated software has been delivered.

The correction program (otccov0000m000) has to be adapted if there are changes to the data model in these customizations (that is O-VRCs, C-VRCs, B-VRCs and possibly L-VRCs), before it can be run for the test company and the test company can be used! See Chapter 1 section 1.3.2, “Customizations”.

One single test company and a number of users are migrated to the new test package combination, for testing purposes (and, possibly, for changes in own software by the customer). In this phase the correction program (tccov0000m000) has to be adapted to the customizations (if necessary), so that the operational conversion can be run without problems.

After testing and acceptance (and possibly adaptation and a second test conversion), the operational companies can be converted to BAAN IV.

Discuss everything with the customer: the backup, TRITON downtime/time frame, the need to adapt tccov0000m000 to changes in the Data Model, and the configuration after the update (test package combination, user(s), and test company/companies to be migrated).

4.2.7 Notes

During installation a new porting set is installed. This implies that all adjustments in directory **$BSE/bin** in their current versions will be lost after the update. The old scripts will still be available (in **$BSE/bin**, under their original name followed by a hyphen [–]).

Make sure that, after the update, nonstandard scripts or scripts subject to change are properly restored or adjusted.

Because the entire environment (TRITON 3 and BAAN IV) is upgraded to Tools version B40_*release*, there will be the following restrictions for the TRITON 3 environment after update:

- When using a graphical user interface (bx or bw) you can not use the zoom (<Ctrl>[Z]) options in an application session to a tools subsession.
- It is not possible to dump run time TRITON 3 forms with BAAN IV tools or to compile TRITON 3 program/report scripts with BAAN IV tools. This implies that after the (software) conversion, the operational TRITON 3 environment will be frozen.
- You must copy **$BSE/tools/tt6.*X_X*/ottadv/oadvstd2** to a backup directory.
- If TRITON 3.0 is running on Oracle, you must create the file **$BSE/lib/defaults/db_resource**, containing the following line:

  ora_level1:1

4.3 Installation procedure

4.3.1 Step 1: Reading out tapes

Read (each of) the tape(s) into a (new) empty directory.

On a suitable file system, create a temporary, empty directory (*baan*, for example) to read the installation tape into. This directory may NOT be below **$BSE**.

Read the tape for C-VRCs (if available) into another directory and the tape for B-VRCs (if available) in yet another.

A. Regular tape

Go to the empty directory and read the tape using the **cpio -icuvdB </dev/***device* command.

B. Master tape

Go to the empty directory and read the first block of the tape (see also the TRITON Tools 6.1 Installation Manual (5966US), Chapter 4, pages 12-32).

When the first block is read, a script called **mas.read6.1** is copied to disk. Start this script. Select the package VRCs and languages to be installed. The script takes care of reading these from tape.

4.3.2 Step 2: Start installation script

The user must be **root**, while the installation is running, the customer cannot use this environment. Check if everyone has left the bshell and has been advised (through the system manager) that TRITON cannot be used during the update.

Check at least the tables for company 000 and also the test company/companies before you start the conversion. You can do this by using the Reorganize tables (ttaad4225m000) session.

Use is made of the (path) data in **$BSE/lib/install/install.data**. It is advisable to check them before installation.

First run the **rc.stop** script and then the **rc.start** script to remove any hanging processes and database locks. If **rc.stop** fails, shut down the environment manually.

Log in as **root** (do not use the **su** command). Check your TERM variable (**echo $TERM**). The vt220 emulation cannot be used: use **vt100** instead.

Before you start the **install6.1** script, you must check if the object oiex1288 resides in the directory **line.000/tttiex**. If not, you must create this directory and copy oiex1288 from the install directory into it.

Go to the installation directory and start the **install6.1** script(via./**install6.1**). The script only prompts you for the **$BSE** directory (enter the **$BSE** directory of the TRITON 3 environment to be updated).

Upon reading the question “Full update?” enter **Yes** (mandatory).

The installation script first calls ottiex1287, then calls reconfiguring the tools tables, and then calls ottiex1288. Under bisam, it takes about 30 minutes to reach this point.

Once the installation script reaches ottiex1288, the rest of the installation will probably run smoothly.

4.3.3 Step 3: Running preinst during or right after the installation of BAAN IV

During an installation or upgrade of the BAAN IV software on a UNIX platform, the **install6.1** installation script calls the **preinst** utility.

The **preinst** utility is part of the BAAN IV installation process, and is used to perform different checks on the UNIX environment. The checks performed by **preinst** concern:

- Kernel parameters
- Checking singlebyte/multibyte database
- Disk space
- Maximum file size
- Required standard users
- TCP/IP configuration
- Environment variables
- Runtime environment versus data dictionary
- Options to speeding up installation
- Install data
- Checking tables
- Customization VRC
- Client/server installation
- Existence of fixes

The **preinst** utility uses the values in the file **kernel_cfg** to check the items listed above. The **kernel_cfg** file contains the minimal kernel parameters of the system, and is installed in the directory **$BSE/lib**. If **preinst** detects that the current value of a parameter is less than the value defined in the **kernel_cfg** file, a warning is displayed. After having finished all the checks, **preinst** displays the following message (if one or more errors are detected):

```
Errors found during system check, continue?
```

You can choose one of the following options:

1. Cancel the installation of the BAAN IV software and tune the UNIX operating system before you restart the installation of BAAN IV.
2. Continue the installation of the BAAN IV software and tune the UNIX operating system after completing the installation of BAAN IV. You do not have to write down the warnings generated by **preinst**, because those messages are written into the **$BSE/log/log.preinst** file.

Note:

During the execution of **preinst** you are asked if you want to check all BAAN IV tables (including those of the runtime environment), or only those of the data dictionary.

The **$BSE/lib/kernel_cfg** file will be set up as shown below. The name of each parameter is followed by a formula to extract its current value, an operator, and a condition.

```
# Kernel configuration file
# Version 6.1
# rklip, 23-09-1996
# Updated for HPUX10.0 and IBM AIX4.1
# Entries that must exist:
# Machine-id
# OS-release
# Kernel
# Kernel type
# MEMORY
# USERS
# MAXFSIZE output in Mb's
# CMD df   <device name> <size in Mb's> <filesystem>
#                                                                                 don't use in the
df command definition awk with printf!
# CMD ping should give percentage packet loss, if not don't
#                                                                                 define the
command
Machine-id: HP9000_PA_RISC
OS-release: HPUX9.0
Kernel: /hp-ux
# Configuration files in order of checking first
Kernel configuration files: /etc/conf/gen/S800, /etc/master
# Kernel type: 1 = static, 2 = static/patch, 3 = dynamic
Kernel type: 1
# Internal memory
MEMORY:
CURRENT: `/etc/sysdef 2>/dev/null | grep "physical memory is" \
           | awk '{print $4}'`
OPERATOR:
CONDITION:
# Number of Triton users
USERS:
CURRENT: `ksh -c "if [ ! -f /.brand/brand6.1 -o ! -f ${BSE}/bin/licmon6.1 ]; \
           then echo 2; else licmon6.1 -b | \
           grep users | awk '{TOT += $4}END{print TOT}'; fi"`
OPERATOR:
CONDITION:
# nproc
NPROC:
CURRENT: `/etc/sysdef 2>/dev/null | grep -i nproc | cut -d= -f 2`
OPERATOR: ge
CONDITION: 64 * USERS
# maxuproc
MAXUPROC:
CURRENT: `/etc/sysdef 2>/dev/null | grep -i maxuprc | cut -d= -f 2`
OPERATOR: ge
CONDITION: 64
...
...
```

4.3.4 Step 4: Check installation logs

After installation, the installation directory should be empty.

When a regular tape is used, the directory **extra** remains.

The log files are in directory **$BSE/log**. Check the most recent ones (type **ls -ltr $BSE/log** to find them). Create a file called **setbse** in **$BSE** that sets the **BSE, BSE_TMP, PATH** and **TERM**(=**vt100**) variables, if it does not exist already.

If the installation has been completed successfully, the customer can start using TRITON again.

4.3.5 Step 5: Import the VRC containing the software components for conversion

To import the VRC containing the software components for conversion, take the following steps:

1 Use the import Data Dictionary session (ttiex1285m000) to import the VRC containing the conversion software into the newest standard VRC (for example, tcB40_b2).

Note:

The name of the VRC containing the conversion software is printed in the letter attached with the delivery tape.

Supply the following information:

Path Sequential Dumps	: delivery directory of conversion software
Path Runtime Dictionary	: $BSE/application
Overwrite	: Yes
Write software components in other Package VRC	: Yes
Package VRC	: newest standard VRC

2 Use the Create Runtime Data Dictionary session (ttadv5210m000) to dump all Package Combination containing the newest standard VRC.

4.3.6 Step 6: B/C - Import customization (ttiex1285m000)

From here on user must be **bsp.** If available, import the customization dumps (B-VRCs and/or C-VRCs) with this session (do not rename them, by importing them into another VRC, because of maintenance reasons). Import the B-VRCs first (if available).

4.3.7 Step 7: Create new VRCs (ttadv1111m000)

The user must be **bsp.** For general rules on the derived-from structure, see document 1034US, page 7.

Create the O-VRCs and U-VRCs using the session:

13 Application Development

3 Packages and Modules

3 Maintain Package VRCs

At least create U-VRCs, derived from the standard B40, for all packages in the format: *package code*/**B40U**/*release*/**stnd/Updates** *package name* **B40_***release*

(example: tc/B40U/b/stnd/Updates Common B40_b).

Use the OPER_001 derived-from structure as a model for TMP_001.

For instace, if the localization (B40L_b_nl10) is available, then for tf and tp derive the U-VRC from the L-VRC instead of from the standard.

Do not include the ti (Business Organizer - B40L_b_org1) localization in the derived-from structure.

In the Business Organizer, items, for instance, are renamed as business objects. Therefore do not incorporate them in an operational/design/testing package combination. For those who want to use this localization, a separate package combination can be created.

Create customer-specific O-VRCs for all packages (exept tt). Basically, copy the existing structure of own VRCs to BAAN IV. Retain the existing customer extension(s) as much as possible.

Example:

OPER_001 contains *3.1O_a_kl01*, derived from *3.1_a1*.
DSGN_001 contains *3.1O_a_dsg1*, derived from *3.1O_a_kl01*.
Now create two new series of own VRCs:
B40O_b_kl01 (derived from *B40U_b_stnd*, as no customization is involved) and:
B40O_b_dsg1 (derived from *B40O_b_kl01*).

4.3.8 Step 8: Create directories for new VRCs (ttadv1115m000)

Create directories for all new (that is U and O) VRCs:

13 Application Development

3 Packages and Modules

4 Maintain Software Component Directories

Press [S] and enter the name of the VRC, press [I] to enter the six software component types. Do this for all VRCs. If the components are entered in the order of the enum (press the spacebar), you can use the TRITON copying facility: Enter the six components for the first VRC. Press <Esc> to go to the choice field, press <Ctrl>[W], [Y], [B], [Y], enter the name of the next VRC, press [Y] and repeat these steps (from <Ctrl>[W] on) for all VRCs.

4.3.9 Step 9: (Own) Copying own work to new own VRCs (ttadv6265m000)

This session has only to be run if there are own developments (mostly menus, forms, or reports). Start the copying session:

13 Application Development

16 Utilities

1 Copy Software Components to New Package VRC

For all own VRCs, copy everything from the old to the proper new O-VRCs.

Also run this session for VRCs that are empty at UNIX level. There may be domains attached to these, or recent developments that are already present in tools may not have been converted to run time.

After this session the new O-VRCs at UNIX level are still (largely) empty; copying is currently only done at tools level.

4.3.10 Step 10: Create New Package Combination(s) (ttaad1120m000)

Check the derived-from structure via the session ttadv1411m000 before using this session.

Call the following session:

2 Application Configuration

8 Maintain Package Combinations

Create at least one new package combination with the name TMP_001 for testing purposes (do not put this in shared memory).

4.3.11 Step 11: Fill Package Combination(s) with VRCs (ttaad1221m000)

Start the session:

13 Application Development

3 Packages and modules

6 Maintain Packages by Package Combination

Fill the new TMP_001 package combination with the right O-VRCs (corresponding with the VRCs in OPER_001 in the TRITON 3 environment). Again: keep to the original structure, but leave out ge (General) and tb (TRITON Support) if available.

End by pressing [Y], to dump the package combination. In BAAN IV runtime Help is created automatically when you use this session to dump a package combination. Therefore, it may take up to an hour before the session is finished.

4.3.12 Step 12: Migrate company/companies (ttaad1101m000)

Convert at least one test company to the new package combination via:

2 Application Configuration

Press [→] twice, to reach the 'Miscellaneous' menu.

2 Change Package Combination by Company

This session may result in a list of warnings (of the type 'Warning: No DD for tcedi472 found'). They relate to new tables and can be ignored.

Run the Check Tables (ttaad4232m000) session now for the test company/companies to make sure that all tables are in the correct format.

4.3.13 Step 13: Migrate user(s) (ttaad2200m000)

Migrate at least one user to the test package combination via:

3 User Management

1 General User Data

2 Change Package Combination by User

4.3.14 Step 14: (Own) Dump forms and menus (ttadv5214m000)

Use this session to convert the forms and menus associated with the migrated own work (if available) to run time:

13 Application Development

15 Run time Data-Dictionary

4 Create Run time Data Dictionary (forms and menus)

Set the Dump field to 1 (default).

4.3.15 Step 15: (Own) Compile reports (ttadv3240m000)

Use this session to compile the reports associated with the migrated own work (if available):

13 Application Development

10 Reports

2 Compile Reports

Set the Compile field to 1 (default). During compilation an error report is generated. Check this report. All warnings can be ignored.

If there are any errors, the normal procedure is to check if they also occurred in the old version (by compiling the old own work via the old package combination, using the same session). If the errors are identical in both versions, there is a problem with the original functionality.

However, this is not possible now because the TRITON 3 environment's tools are now of the version B40_*release.*

4.3.16 Step 16: Validate the system

Get a printout of the commercial (tools) functions the customer has bought. Validation of the toolset can only take place when logging on to the customer system as **root** (or using the **su** command). Do not forget to set the environment variable by running **./setbse**. After the validation process, log on as **bsp** again and start the bshell.

Important note:

Never give the Baan default password to customers.

Choose menu options:

6 Maintain Security Code/Validation Key (ttiex3102m000)

Enter the customers data from the printout of commercial information for this customer. For Baan NL customers, place a 2 before the customer number. All the other information can be found on the customer data printout.

2 Maintain Requested System Configuration (Tools) (ttiex3101m000)

Type [S] to enter the correct date. Then type [C] to edit the screen. Enter the data provided in the commercial printout. Do not make any changes.

1 Maintain Requested System Configuration (ttiex3100m000)

For BAAN IVa enter **Yes** for all entries, for all packages. Press [N] to go to the next page; <Ctrl>[N] to go to the next package.

For BAAN IVb only special items (t103 Development Tools) have to be set to **Yes** when they are bought and not the entire software configuration (as in BAAN IVa).

3 Print Requested System Configuration ()

Press [Y]. Enter: [D] (Display) or ASCIF (ASCII file; by default this file will be placed in the current directory). In the latter case, choose a logical file name.

If applicable, read in the file via a subshell ([<]).

Provide this printout to Baan. This will be used to create the validation key. When you receive this key in a printout from Baan, proceed with the validation.
Choose menu options:

1 Software Installation

6 Maintain Security Code/Validation Key (ttiex3102m000)

Press [C] to edit the screen; enter the Validation Key (Tools) and the Validation Key Software (Standard). Make sure the Security Key on the Baan printout is exactly the same as the one in this session. Press [E] to exit.

Note:

To run the next session, user has to be **root**. Change back to **bsp** after completing Validate Toolset and starting the license-daemon.

7 Validate Toolset (ttiex3220m000)

To the question about starting the license server, answer Yes. At this point an error message may appear. If a message appears, check if the license daemon is running.

You may have to start the license daemon manually. The license daemon must be running in order to complete the next step (otherwise another error message appears).

Check the license daemon by leaving the bshell (via [<] at the Choice prompt) and by entering ps -ef | grep licd. The license daemon is a process with the name licd6.1. If necessary, start the license daemon by entering the licd6.1 command. Return to your bshell (type exit) and perform the following action.

8 Configure Application (ttiex3225m000)

This is also known as patching the software. Enter: **Yes** (the next field, **No**, is skipped), **Yes**. Press [Y] to confirm. If you see Sorting... message, the process has been started correctly.

4.3.17 Step 17: Create the audit database and assign tables to it

BAAN IV includes an audit facility that can log all parameter changes. To use this audit, you must first create an audit database and assign tables to it. To do so, follow the steps described below:

1. Log on as user **bsp**.
2. Select the Database Management menu from the Tools main menu.
3. Select the Maintain Database Definitions (ttaad4110m000) session from the Maintenance menu.
4. Insert a new database named AUDIT. As description, use **AUDIT Database**. The type of database depends on the running RDBMS. Select the Audit Trail check box. Clear the Force to Read and Copies Present check boxes.
5. Select the Maintain Audit File Directories (ttaad4116m000) session from the Maintenance menu.
6. Select All (other) tables in the Table Selection field and insert the path name for the audit directory, for example, **/usr3/audit /#** (the number sign indicates that for each company a separate directory is created under **/usr3/audit**).
7. Select the Convert to Runtime option from the Application menu.
8. Select the Database Management menu from the Tools main menu.

9 Select the Assign Tables to Databases (ttaad4111m000) session from the Maintenance menu.

10 Insert all tables you want to be audited. Note that the database field must be filled with the AUDIT database. The Table Selection field enables you to specify individual tables, all tables within certain modules, and all (other) tables. You can insert the mandatory tables by choosing the Load Mandatory Tables for Audit option from the Application menu.

11 Select the Convert to Runtime option from the Application menu and close and restart the **bshell**.

4.3.18 Step 18: Create tables (ttaad4230m000)

For all test company numbers migrated to BAAN IV, create new empty tables using the session:

6 Database Management

Press [→] twice, to reach the Miscellaneous menu.

3 Create Tables

4.3.19 Step 19: Parameters (tcmcs0295m000)

Run this session for all test company numbers that were migrated to BAAN IV.

4.3.20 Step 20: Maintain Company Parameters (tfgld0103m000)

Run this session to insert the appropriate Group Company number.

4.3.21 Step 21: Maintain ITM Parameters (tiitm0100m000)

Run this session to insert the appropriate Prefix for Item Code Container Items.

4.3.22 Step 22: Maintain Base Units (tcmcs0100m000)

Run this session to insert the appropriate base units.

4.3.23 Step 23: Checking and finalizing

Perform the following checks and actions:

- Make sure the scripts in **$BSE/bin** are restored to their old state.
- Select the Software Installation menu from the main menu, and start the Modify Codes of Alignment session in the Miscellaneous menu. Make sure to align the articles the same as in the old installation. If you do not do so, the conversion will fail.
- Define the proper parameters for Common by using the session tcmcs0100m000.
- Define the proper parameters for ITM by using the session tiitm0100m000.

Now turn to the next chapter for performing the final step: running the correction program for the test company/companies.

5 Correction program and checking the conversion: both scenarios

This chapter describes the conversion process of TRITON version 3 to BAAN IV. It also supplies a conversion matrix, in which the different conversion programs used for the TRITON/BAAN IV packages are listed.

The second section of this chapter describes the correction program that has to be run (for both scenarios) for the test companies.

Note that the Conversion of TRITON 3.x => BAAN IV (tccov0000m000) session has to be adapted to the possibly customized data model of BAAN IV, in case this is not standard (see section 1.3.2).

The third section describes the sessions needed by package to perform a general and a specific data check after the conversion has taken place.

Note:

The procedure described in section 5.5.4, Data conversion from BAAN IVa to BAAN IVb, must always be followed, even if you are converting a TRITON 3.x version to BAAN IVb.

5.1 Conversion matrix release 3 release 4

This section contains a matrix, that gives an insight in which conversion programs are used converting to BAAN IVb, coming from release 3.

Note the following points:

- There is no conversion required for the extensions (sch, nl1) that are not mentioned in the table below.
- There is no conversion required for the packages (tt, tc, cp ps) that are not mentioned in the table below.
- No conversion is required for T3.x extension CMS (import/export).
- All package related conversion programs are included and automatically executed by conversion program **tccov0000m000**.

Release	TI	TD	TR	TF	TU	TP	TS	glo/gls
T30b	tccov3001 tccov3002 tccov3003 tccov3004	tccov1001 tccov1002	tccov5001	tccov2001 tccov2002 tccov2003 tccov2004 tccov2005 tccov2006 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002	exchange	tfcmgl900
T30b1	tccov3001 tccov3002 tccov3003 tccov3004	tccov1001 tccov1002	tccov5001	tccov2003 tccov2004 tccov2005 tccov2006 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002	exchange	tfcmgl900
T31a	tccov3002 tccov3003 tccov3004	tccov1002	tccov5001	tccov2001 tccov2002 tccov2003 tccov2004 tccov2005 tccov2006 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002		tfcmgl900
T31a1	tccov3002 tccov3003 tccov3004	tccov1002	tccov5001	tccov2003 tccov2004 tccov2005 tccov2006 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002		tfcmgl900
T31b	tccov3002 tccov3003 tccov3004	tccov1002	tccov5001	tccov2006 tfcor0300 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002		tfcmgl900
T31b2	tccov3002 tccov3003 tccov3004	tccov1002	tccov5001	tccov2006 tfcor0300 tccov3005 otfcor0260 otfcor0261	tuxch9211	tccov3002		tfcmgl900
B4a				tccov3005 otfcor0260 otfcor0261				tccovl999 (optional)

5.2 Conversion of TRITON 3.x => BAAN IV (tccov0000m000)

Basically, this session is a shell around a number of correction and conversion sessions which are used to convert the data of an older TRITON 3.x version to BAAN IVa.

First, the right TRITON version must be selected as the original version. Next the packages which must be converted to BAAN IVa must be selected since this session will only convert and correct the data of selected packages. On the second form of this session, the current status of each correction or conversion part is displayed. Next, the Real Conversion field, which is the most important parameter in this session, must be set to **No.**

Now you can run a test conversion which reports all changes that will occur as a result of a conversion. Run test conversions until the conversion is executed correctly. This is done to ensure that the real conversion will be successful.

At the bottom of the form you can find information regarding the version which is converted, the user who performed the conversion, and the number of times the conversion has been started is displayed.

Note:

Log files are created in the directory **$BSE/log**, containing the output of the conversion sessions. These log files must be checked. For an explanation of the output of the conversion sessions, refer to Appendix B.

For the different packages, the following sessions are executed to correct or convert the tables to BAAN IV (if applicable):

5.2.1 BAAN Distribution

tccov1001m000 (conversion of distribution 3.X => 3.1B)
By using this session, the following tables are converted:

TRITON Distribution (Sales and Marketing Info)

- Selections
  The Expressions from table Selection Lines are copied to Selection Expression of Selections for the corresponding Selections. The field tdsmi070.expr will be filled in with tdsmi071.snse for all corresponding selection lines.
- Letters
  The Layouts from table Layouts by Letters are copied to Layout of Letters for the corresponding Letter. The field tdsmi150.text will be filled in with tdsmi151.layo for all corresponding layout lines.
- Planned INV Purchase Orders
  The Company from Planned INV Purchase Orders is filled in with the default Company for the supplier. For single financial companies, this field is filled in with the default financial company for the logistic company. The field tdinv310.comp will be filled in with the default company number from the supplier.
- Purchase Order/Receipts
  The Financial Company from the Purchase Order Header is copied to the Purchase Receipts. The field tdpur045.fico will be filled in with tdpur040.comp.
- Sales Orders
  The Financial Customer Group is filled in with the Financial Customer Group of the customer of the Sales Order. The field tdsls040.cfcg will be filled in with tccom010.cfcg for the corresponding tdsls040.cuno.

TRITON Common:

- Messages to be generated
  The Parent Relation is filled in with the Relation. The field tcedi700.pren will be filled in with tcedi700.reno.
- Generated Batch Numbers
  The table Generated Batch Numbers is filled in with the corresponding data from Generated Messages. The following fields will be converted:
  - tcedi720.netw = tcedi701.netw
  - tcedi720.bano = tcedi701.bano
  - tcedi720.pren = tcedi701.reno
  - tcedi720.send = tcedi701.send
  - tcedi720.sent = tcedi701.sent
  - tcedi720.stat = tcedi701.stat
  - tcedi720.lnst = tcedi701.mstt

tccov1002m000(conversion of distribution 3.1B => BAAN IV)
By using this session, the following tables are converted:

Parameter Tables

The following parameter tables are created:

tdinv000	Inventory Parameters
tdilc000	Location Control Parameters
tdltc000	Lot Control Parameters
tdpur000	Purchase Control Parameters
tdsls000	Sales Control Parameters
tcedi000	Electronic Data Interchange Parameters

TRITON Common

- EDI Messages
  The fields Session Incoming Message and the Session Outgoing Message are converted to Library Incoming and Library Outgoing. The field tcedi005.sesi will be filled in with the first five positions of the contents of the source field + **dll** + the last four positions of the contents of the source field. The field tcedi005.seso will be filled in with the first five positions of the contents of the source field + **dll** + last four positions of the contents of the source field.

Note: The target field will be empty if the source field was empty.

Purchase Control

- Purchase Inquiry Lines/Purchase Inquiry Line History
  The Item Category is filled in with ‘Customized’ for Inquiry Lines for a project, and with ‘Standard’ for the others. The field tdpur002.opol and tdpur011.opol will be filled in with tcittb.customized if tdpur002.cprj or tdpur011.cprj is filled in.
- Purchase Discount Parameters by Supplier and Price Group
  For Grade by Quantity/Amount equal to Quantity, the Unit is filled in with the Price Unit from the Price Group when the Price Group is filled in. The field tdpur034.cuni will be filled in with tcmcs024.cpru if the field tdpur034.cpgp is filled in.

Note: The tdpur034.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Purchase Discount Parameters by Price List and Price Group
  For Grade by Quantity/Amount equal to Quantity, the Unit will be filled in with the Price Unit from the Price List when the Purchase Price List is filled in. The field tdpur035.cuni will be filled in with tcmcs024.cpru for the corresponding price list, provided that the price list is not empty.

Note: The tdpur035.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Purchase Price Parameters by Supplier and Item
  The Unit (when empty) is filled in with the Purchase Price Unit from the Item table (tiitm001), or when this field is empty, the Inventory Unit. If tdpur036.cuni is empty, then it will be filled in with tiitm001.cupp. If tiitm001.cupp is emty, then tdpur036.cuni will be filled in with tiitm001.cuni.

Note: The tdpur036.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Purchase Price parameters by Price List and Item
  The Unit (when empty) is filled in with the Purchase Price Unit from the Item table (tiitm001), or when this field is empty, the Inventory Unit. If tdpur037.cuni is empty, then it will be filled in with tiitm001.cupp. If tiitm001.cupp is empty, then tdpur037.cuni will be filled in with tiitm001.cuni.

Note: The tdpur037.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

Sales Control

- Sales Quotation Lines
  The Success Percentage is copied from the Sales Quotation Header.
  The Item Category is filled in with ‘Customized’ for Sales Quotation Lines for a Budget.
  The field tdsls002.fact will be filled in with the tdsls001.fact. The field tdsls002.opol will be filled in with tcittc.customized if the tdsls002.cbdg is not empty.

- Sales Quotation Line History
  The Item Category is filled in with ‘Customized’ for Sales Quotation Lines for a Budget.
  The Success Percentage is copied from the Sales Quotation History.
  For Sales Quotation Lines in History with Record Type (1-4) equal to 3, the Status is filled in with ‘Canceled’. For Sales Quotation Lines in History with Record Type (1-4) equal to 4, the Status is filled in with ‘Processed’, and Reason for Blocking and Competitor are copied from the corresponding Sales Quotation History (Header). The field tdsls011.opol will be filled in with tcittc.customized if the fielde tdsls011.cbdg is not empty. If the field tdsls011.ckor is equal to 3, then tdsls011.lsta will be filled in with tdsls.lsta.abandoned. If the field tdsls011.ckor is equal to 4, then tdsls011.lsta will be filled in with tdsls.lsta.processed, tdsls011.crea will be filled in with tdsls010.crea, and tdsls011.ccom will be filled in with tdsls010.ccom. The field tdsls011.fact will be filled in with the tdsls010.fact.

- Sales Discount Parameters by Customer and Price Group
  For Grade by Quantity/Amount equal to Quantity, the Unit is filled in with the Price Unit from the Price Group when the Price Group is filled in. The field tdsls034.cuni will be filled in with tcmcs024.cpru if the tdsls034.cpgp is filled in.

Note: The tdsls034.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Sales Discount Parameters by Price List and Price Group
  For Grade by Quantity/Amount equal to Quantity, the Unit is filled in with the Price Unit from the Price List when the Sales Price List is filled in. The field tdsls035.cuni will be filled in with tcmcs024.cpru for the corresponding price list, on condition that the price list is not empty.

Note: The tdsls035.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Sales Price Parameters by Customer per Item
  The Unit (when empty) is filled in with the Sales Price Unit from the Item table (tiitm001), or when this field is empty, the Inventory Unit. If tdsls036.cuni is empty, then it will be filled in with tiitm001.cupp. If tiitm001.cupp is emty, then tdsls036.cuni will be filled in with tiitm001.cuni.

Note: The tdsls036.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Sales Price Parameters by Price List and Item
  The Unit (when empty) is filled in with the Sales Price Unit from the Item table (tiitm001), or when this field is empty, the Inventory Unit. If tdsls037.cuni is empty, then it will be filled in with tiitm001.cupp. If tiitm001.cupp is empty, then tdsls037.cuni will be filled in with tiitm001.cuni.

Note: The tdsls037.kogr must be equal to tckogr.quantity, otherwise the field will not be converted.

- Sales Deliveries
  The Currency Rate Sales and the Rate Factor are copied from the Order Header. The field tdsls045.ratf will be filled in with tdsls040.ratf, tdsls045.rats with tdsls040.rats, and tdsls045.pmsk will be 20 positions instead the 15 positions in the source field (the 5 new positions will be **nnnnn**).
- Sales Order Line History
  The Currency Rate Sales and the Rate Factor are copied from the Order Header. The field tdsls051.ratf will be filled in with tdsls050.ratf and tdsls051.rats with tdsls050.rats.

Inventory Control

- Planned INV Production Orders/Planned INV Purchase Orders
  The Planner is copied from the item table (tiim001) for the order's item. The fields tdinv300.cplb and tdinv310.cplb will be filled in with the tiitm001.cplb.

Replenishment Order Control

- Replenishment Order Types
  The Order Procedure is filled in with the Replenishment Order Procedure from the TOC Parameters (trtoc000). The field tdrpl004.cpro will be filled in with trtoc000.cprr.

  The Order Series are filled in with the Order Series from the linked Order Procedure (trtoc030). The field tdrpl004.orse will be filled in with trtoc030.orse. The link is done via tdrpl004.cpro.

TRITON Common

- Order Types
  The Series are filled in with the Order Series from the linked Order Types (trtoc030). The field tcmcs042.cpro will be filled in with trtoc000.cprs on condition that tcmcs042.modc is equal to tcmodc.sls. Else, tcmcs042.cpro will be filled in with trtoc000.cprp.
  Next the Order Procedure is filled in with Purchase or Sales Order Procedure (dependent of the value for 'Purchase/Sales') from the TOC Parameters (trtoc000). The field tcmcs042.orse will be filled in with trtoc030.orse. The link is done via tcmcs042.cpro.
- COM Parameters
  Table (tccom999) will be filled in with the corresponding fields from the Parameters (tcmcs095). The following fields will be converted:
  - tccom999.crtf
  - tccom999.bnch
  - tccom999.trtr
  - tccom999.rttr
  - tccom999.aust.7
  - tccom999.ecvr.7
  - tccom999.sitm.7
  - tccom999.lsci.7
- Order Types
  The blocking Order Step is copied from the Parameters (tcmcs095). The blocking order step from tcmcs095 will be copied to all records in tcmcs042.blcs.

Procedure Masks

- The Procedure Mask for each Order Type (Purchase, Sales, Replenishment), Purchase Delivery and Replenishment Order Line is extended from 15 positions to 20 with **nnnnn**. The field tcmcs042.pmsk will be filled in with 15 times **n**.

Purchase and Sales Statistics

- The Primary Key Field is changed
  The domain length of the fields, which compose the primary key field is changed. The following tables are converted:
  - tdpst010: PURCHASE Statistics Master File (Turnover)
  - tdpst015: PURCHASE Statistics Master File (Order Intake)
  - tdpst018: PURCHASE Statistics Master File (Cancellations)
  - tdpst020: PURCHASE Statistics Compression (Turnover)
  - tdpst025: PURCHASE Statistics Compression (Order Intake)
  - tdpst028: PURCHASE Statistics Compression (Cancellations)
  - tdpst030: PURCHASE Statistics Sort File (Key File)
  - tdpst050: PURCHASE Budgets
  - tdsst010: SALES Statistics Master File (Turnover)
  - tdsst015: SALES Statistics Master File (Order Intake)
  - tdsst018: SALES Statistics Master File (Cancellations)
  - tdsst020: SALES Statistics Compression (Turnover)
  - tdsst025: SALES Statistics Compression (Order Intake)
  - tdsst028: SALES Statistics Compression (Cancellations)
  - tdsst030: SALES Statistics Sort File (Key File)
  - tdsst050: SALES Budgets

5.2.2 BAAN Manufacturing

- Conversion of Manufacturing 3.1A => BAAN IV (tccov3001m000)
  By using this session, a number of fields and tables are modified, namely:
  - The surcharge history is created based on the following tables:

    Cost Price Surcharges by Item and Cost Price Component (ticpr220)
    Cost Price Surcharge Bases by Item and Cost Price Component (ticpr230)
    Cost Price Surcharges by Item and Effective Date (ticpr320)
    Cost Price Surcharge Bases by Effective Date (ticpr330)

Note: This will only be done if the parameter CPR, 3101 is not present in Statuses.

  - The value of the Drawing field in the Engineering Item Revisions table is deleted. The drawing code can be empty, because the user did not have the functionality before.
  - The ECO Lines table is set to the initial value of the data definition. For all records in (ECO Lines) tiedm320 field (Replace Type) tiedm320.korp will become multiplication (tiedm.korp.multiplication).
  - The Device for Messages in Generate MPS parameter is set to the initial value of the data definition.
  - The Only Reschedule Messages for Total Order Quantity parameter is set to the initial value in the data definition.
  - Effective Data of the Bill of Critical Capacity is set to 01/01/1990 if the date field was left empty.
  - The value of the Finance Company field in the Planned MPS Interplant Orders, Planned MPS Purchase Orders, and Planned MRP Purchase Orders tables is entered.
  - The value of the Financial Supplier Group field in the Financial Project Transactions table is deleted.
  - The horizon for the company calendar will be set to 50 days in the past and 500 days into the future.
- Conversion Units to Units by Unit Sets (tccov3002m000)
  Because of the new way in which (alternative) units are handled, the following tables are converted and modified by means of this session:

Table	Fields used for creating unit sets	Field filled in
Engineering Items	tiedm010.cuni	uset
Items	tiitm001.cuni, stgu, cuqp, cupp, cuqs, cups, cntr	uset
Alternative Units	tiitm003.cuni	-
Item Default Data	tiitm002.cuni, stgu, cuqp, cupp, cuqs, cups, cups	uset
Calculation Parts	tipcs002.cuni	uset
Customized Items	tipcs021.cuni, stgu, cuqp, cupp	uset
Equipment	tppdm025.cuni	uset
Subcontracting	tppdm035.cuni	uset
Alternative units by project item	tppdm604.cuni	-
Project items	tppdm605.cuni, cuqp, cupp	uset
Project Equipment	-	tppdm625.uset
Project subcontracting	tppdm635.cuni, cuti	uset

Note:

Unit sets (tcmcs006) will be created and the corresponding unit set in the tables will be filled in. Also the table unit by unit set will be filled in.(tcmcs012).

- Conversion Unit Conversion Factors (tccov3003m000)
  Because of the new way in which (alternative) units are handled, the following conversion factors are converted and modified by using this session:

  – Tables Units(tcmcs001)
  – Item Table (tiitm001)
  – Alternative Units(tiitm003)
  – Customized Items(tipcs021)

  Table conversion factor (tiitm004) will be filled in based on found conversion factors between different units. Conversion factor will be stored on different levels. (General, Item Group, Item).

  The tables Units, Alternative Units, and Customized Items must remain unchanged in the BAAN IV version, as in 3.1, which has the following two advantages:

  – The conversion can be performed after the installation of BAAN IV
  – The conversion can be performed anew

  The conversion of conversion factors may involve the following conflicting situation:

  – In versions 3.1 and lower it was possible to record any conversion factor. In BAAN IV, however, it will not be possible to record a (customized) item-specific conversion factor for conversion between two units (except for units with the same physical quantities). The session will recognize these conflicting situations and write them to the log file. However, if a final conversion is performed, a different conversion factor will be generated.

- Fill Auxiliary Item Table (tiitm100) tccov3004m000)
  With the introduction of item containers in BAAN IV, this session is used to ensure that the number of records in the tiitm100 table is the same as in the tiitm001 table.

5.2.3 BAAN Finance

- Change Sign in Open Item Table (tfacp200) (tccov2001m000)
  By means of this session, the symbol for all currency differences of anticipated payments is modified. Note that tfacp200.cdam is multiplied with -1.
- Change Step for Antic. Unalloc/Adv Payments (tccov2002m000)
  For anticipated advance, unallocated anticipated or standing orders in tfacp200 the field tfacp200.step will be filled in as follows (depending on the payment method) with the first possible option:
  - When tfcmg003.rs01 = yes then document sent to customer
  - or else if tfcmg003.rs02 = yes then document accepted by customer
  - or else if tfcmg003.rs03 = yes then document sent to bank
  - or else if tfcmg003.ps01 = yes then document received
  - or else if tfcmg003.ps02 = yes then document accepted/sent

  In tfacr200 the same happens for advance anticipated and unallocated anticipated.
- Check Standing Order (tfcmg114) (Posting Data) (tccov2003m000)
  By means of this session, the Standing Order Number field is added to the Posting Data for Anticipated Payments without Invoice table. This program reads the anticipated table and will update tfcmg114. In tf3.1b an extra field for the standing order is added to tfcmg114. This field will be filled in with this program. From tfcmg111 the standing orders are read which are of type cost in tfcmg114. The field tfcmg114.orno will be filled in with tfcmg111.docn.
- Check Anticipated Document (tfacp200/tfacr200) (tccov2004m000)
  By means of this session, the tables for unallocated/advance receipts/payments are checked to ensure that the anticipated document number is not specified. If the anticipated document number is specified, this is corrected after which the selected records are saved in a file. This program reads the open item tables for unallocated/advance receipts/payments. It will check if the anticipated document number is filled in. For unallocated/advance payments it should not be filled in. If these fields are filled in it will cause differences on the checklist (tfacp2415/tfacr2415) for both the unallocated and anticipated document. The total difference of these documents will be 0. Note the following points:
  - Also tfgld102 and tfgld106 will be updated.
  - In tfacp200 both tfacp200.typa and tfacp200.doca are emptied for unallocated and advance. The same applies in tfacr200 for typa and doca.
  - In tfgld102 both 102.ctyp and 102.codc are emptied for documents of category cash which are also present in tfcmg110. The same happens in tfgld106 for the fields ctyp and cinv.
- Change Enumerated Field (tfcmg108) (tccov2005m000)
  By means of this session, the number of enum values of a field in the One Time Supplier Addresses table is extended from 2 to 8.

- Update Payment Differences from Anticipated Payment (tccov2006m000)
  In tfcmg111 the invoices, purchase invoices and the sales invoices are read. When an invoice has payment method tfcmg003.repa = payment or when it is a purchae invoice the document from tfcmg111 will be looked up in tfacp200. When it is an anticpated document tfacp200.padi, pada and padh are updated.

  tfacp200.padi:
  If tfcmg111.amnt is greater than 0 and less than tfacp200.amnt, tfacp200.padi will be set equal to the negative value of tfcmg111.pdif.

  If tfcmg111.amnt is less than 0 or greater than tfacp200.amnt, tfacp200.padi will be set equal to tfcmg111.pdif.

  tfacp200.pada and tfacp200.padh will be converted from tfacp200.padi using the currencies from tfcmg111. Sales invoices are converted similarly in tfacr200.

5.2.4 BAAN Transportation

- Conversion of Transportation 3.x => BAAN IV (TCO) (tccov5001m000)
  By means of this sessions, the value of the Match Status cost line field is set to *All Approved* if there is a link with Finance and a cost line with status *Posted*.

Each of these sessions has an indicator which displays the conversion status. The following statuses can be distinguished:

- *Status not defined*
  This means that the initial value is displayed.
- *Execution not necessary*
  This means that the conversion does not apply to the original version and packages which were selected.
- *Not executed*
  This means that the conversion must still be executed.
- *Execution aborted*
  This means that the conversion was aborted due to an error which occurred during the conversion.
- *In process*
  This means that the session is running.
- *Execution completed*
  This means that the conversion was successful.

For each session, a logfile is created in which data regarding the events, the converted components, and the errors which occurred during each conversion. These log files are located in the $BSE/log directory. For example: $BSE/log/log.cov1001.

5.2.5 Converting TRITON Service version 3.0b to BAAN IV

This section describes the conversion of TRITON Service version 3.0b to BAAN IV. This conversion is needed because of the differences between Service version 3.0b and version 3.1a. For this conversion the Exchange package must be used. This section contains the following topics:

- Converting TRITON Service 3.0b to BAAN IV
- Checking the conversion

Converting TRITON Service 3.0b to BAAN IV

Take the steps described below to convert TRITON Service 3.0a to BAAN IV.

1 Execute **bdbpre6.0** for the Services tables of TRITON version 3.0. You must use a shell script that does the following:

- Uses ^A as seperator
- Converts the **t***ppmmmxxxccc***.S** syntax to *ppmmmxxx*. Where *pp* stands for the package code, *mmm* for the module code, and *xxx* for the table number.

2 Create two company numbers in BAAN IV, one of which will be used for Exchange only. These Companies must be linked to the same (dedicated) Packet Combination. Note that this Packet Combination must not be the operational Packet Combination.

3 Use the session ttadv5217m000 to remove the reference integrity from the Package Combination.

4 Install the Exchange Scheme S30b_31a by using the session Create Tables from Sequential Dump (ttaad4227m000) in one of the Companies created in step 2.

5 If there are customizations, run the Copy Exchange Scheme session (tuxch0201m000) to copy the entire Exchange Scheme.

6 Use the Maintain Exchange Scheme (tuxch0101m000) session to define the pathnames for the Exchange objects, Import log file, and Sequential files. This path must refer to the directory, in which the tables dumps of TRITON 3.0 reside.

7 Use the Create Import Scripts (tuxch0227m000) session to compile all condition scripts.

8 Create all BAAN IV tables for the second Company (the company for the coversion results).

9 Enter the Company data for the Company for the conversion results by using the session Maintain Companies (tccom0100m000).

10 Run.the conversion program Import data (tuxch0223m000). The Overrule batch Company option must be set to Yes, and the Company number must be the Company for the Conversion results.

11 If you have customizations, run the session Copy Exchange Scheme (tuxch0201m000) to copy the entire Exchange Scheme.

12 Use the session Create Runtime Data Dictionary (ttadv5120m000) for the Package Combination with Conversion results. The referential integrity will become active.

13 All references must be checked and built up using the session Reorganize Tables (ttaad4225m000).

Checking the conversion

This section only describes the major changes which are essential for the consistency of the new Service environment. To describe all the changes is too time consuming, but you can use session Print Field Relations (Import) (tuxch0422m000) to print the field relations.

The following tables are converted:

- tssma000 (Service Parameters)
  Within this table a lot of new parameters are added for planning purposes. These parameters will get all default values during the conversion. The session Maintain Service Parameters (tssma3100m000; especially form 4) can be used to verify the conversion.
- tssma110 (Locations)
  The major changes is that for the locations the address data is removed from the table. There are two types of addresses introduced for the location: Postal and Work site Addresses. Only for the work site there is a conversion possible. The session Maintain Locations (tssma1110m000) can be used to verify the conversion.
- tssma121 (Items by Servicing)
  For the required items by servicing also the warehouse have to be filled in case of "inventory" items. The session Maintain Items by Servicing (tssma1121m000) can be used to verify the conversion.
- tssma123 (Servicings)
  The location of the installation must be filled during the conversion. The field is not used on a form but is used in the background while generating a maintenance plan. The table maintenance session (ttaad4100) can be used to verify the conversion.
- tssma124 (Periodc Maintenace Planning)
  To plan the maintenace by location and to plan more precise within the maintenace period the location and planned execution date will be filled during the conversion. The sessions Maintain Periodic Maintenance Planning (tssma1124m000) and Maintain Periodic Maintenance Planning by Location (tssma1125m000) can be used to verify the conversion.
- tssma301 (Service Orders)
  The major changes is that for the Service Orders the address data is removed from the table. The session Maintain Service Orders (tssma3101m000) can be used to verify the conversion. There are two types of addresses introduced for the Service Order: Postal and Work site Addresses.

  For the conversion of Worksite the following two situations can occur:

  - When the location on the order is not filled
    The conversion will check if the customer data within the order is the same as the data in the customer table. If the data is the same not specific worksite address will created. When the data differs a specific worksite address is created.
  - The location on the order is filled
    In this case the following can be done:
    - The conversion will check if there is specific location worksite address data, when this differs from the order data a specific worksite address is created.
    - When no specific location address data is specified but only a worksite address code the system will check this data against the order data, when differs a specific worksite address is created.
    - When no worksite address code is and the customer data differs from the order data a specific worksite address is created.

For the conversion of Postal Address the following two situations can occur:

- When the location is not filled the postal address code is copied.
- When a location is filled, the Postal Address data is derived from the location.

5.3 General data check

Although there are no extensive changes in the data structure, it is necessary to check whether or not the data has been converted. This can be done by drawing a number of samples from a number of tables in the different packages.

5.3.1 TRITON Common

TRITON 3.1b	BAAN IV	Description
tccom1401m000	tccom1401m000	Print Customers
tccom2401m000	tccom2401m000	Print Suppliers

5.3.2 TRITON Distribution

TRITON 3.1b	BAAN IV	Description
tdilc1414m000	tdilc1414m000	Print Inventories by Warehouse and Item
tdinv6220m000	tdinv6220m000	Perform Slow-Moving Analysis by Item
tdinv6230m000	tdinv6230m000	Perform Inventory Valuation by Item
tdinv7410m000	tdinv7410m000	Print Stock Transactions by Item
tdinv8450m000	tdinv8450m000	Print Chart of Accounts for Stock and WIP Transactions
tdpst0401m000	tdpst0401m000	Print Purchase Statistics
tdpur4404m000	tdpur4404m000	Print Purchase Invoices
tdpur4407m000	tdpur4407m000	Print Purchase Orders by Order Number
tdpur4431m000	tdpur4431m000	Print Receipts
tdpur5401m000	tdpur5401m000	Print Purchase Order History by Supplier and Item
tdsls1403m000	tdsls1403m000	Print Quotations by Item
tdsls4404m000	tdsls4404m000	Print Sales Invoices
tdsls4410m000	tdsls4410m000	Print Sales Orders by Order Number
tdsls4431m000	tdsls4431m000	Print Sales Order Deliveries
tdsls5401m000	tdsls5401m000	Print Sales Order History by Customer and Item
tdsls6401m000	tdsls6401m000	Print Chart of Accounts - Invoice Analysis
tdsls6402m000	tdsls6402m000	Print Chart of Accounts - Invoice Analysis by VAT Code
tdsls6403m000	tdsls6403m000	Print Chart of Accounts - Invoice Details
tdsst0401m000	tdsst0401m000	Print Sales Statistics

5.3.3 TRITON Manufacturing

TRITON 3.1b	BAAN IV	Description
tibom1214m000	tibom1214m000	Detect Loops in Production BOMs
tibom1411m000	tibom1411m000	Print Production Bills of Material (Multi-Level)
ticpr0410m000	ticpr0410m000	Print Cost Price Components
ticpr2410m000	ticpr2410m000	Print Surcharges by Item Group
ticpr2410m000	ticpr2410m000	Print Cost Price Calculation by Item
ticst0401m000	ticst0401m000	Print Actual Costs by Production Order
ticst2401m000	ticst2401m000	Print Costing History by Order
tiedm1470m000	tiedm1470m000	Print EBOM and PBOM Consistency
tigrt0441m000	tigrt0441m000	Print Classified Items by Family
tihra1401m000	tihra1401m000	Print Hours Accounting by Employee
tihra3460m000	tihra3460m000	Print Hours History by Employee
tiitm0401m000	tiitm0401m000	Print Item Data
tipcf0450m000	tipcf0450m000	Print Product Features
tipcf2401m000	tipcf2401m000	Print Constraint IDs by Generic Item
tipcf3411m000	tipcf3411m000	Print Generic Bill of Material (Multi-Level)
tipcf4401m000	tipcf4401m000	Print Generic Price Lists
tipcf6410m000	tipcf6410m000	Print Product Variant Statistics
tipcs1460m000	tipcs1460m000	Print Cost Price by Budget
tipcs1462m000	tipcs1462m000	Print Cost Price Calculation by Budget
tipcs2451m000	tipcs2451m000	Print Customized Bills of Material (Multi-Level)
tipcs3460m000	tipcs3460m000	Print Cost Price by Project
tipcs3462m000	tipcs3462m000	Print Actual Cost Price by Project
tipcs8410m000	tipcs8410m000	Print Financial Project Transactions
tipcs8450m000	tipcs8450m000	Print Chart of Accounts for Financial Project Transactions.
tisfc0401m000	tisfc0401m000	Print Production Order Report
tisfc0450m000	tisfc0450m000	Print Work-in-Process
tisfc2401m000	tisfc2401m000	Print Subcontracted Operations by Supplier

5.3.4 TRITON Finance

TRITON 3.1b	BAAN IV	Description
tfacp2420m000	tfacp2420m000	Print Aging Analysis by Supplier
tfacp2415m000	tfacp2415m000	Print Control Account Checklist
tfacr2420m000	tfacr2420m000	Print Aging Analysis by Customer
tfacr2415m000	tfacr2415m000	Print Control Account Checklist
tffst1404m000	tffst1404m000	Print Financial Statements
tfgld0411m000	tfgld0411m000	Print Transaction Type Master
tfgld0408m000	tfgld0408m000	Print Chart of Accounts
tfgld1406m000	tfgld1406m000	Print Finalized Transactions by Ledger Account
tfgld3402m000	tfgld3402m000	Print Trial Balance
tfgld3409m000	tfgld3409m000	Print Column Balance
tfgld3405m000	tfgld3405m000	Print Trial Balance with Dimensions

5.4 Specific data check

During the final step in the conversion, a number of sessions are automatically activated (if required) to convert specific data. The lists of sessions below can be used to check the modifications. Read also the session help of the conversion sessions for more information about the modifications.

5.4.1 BAAN Distribution

Conversion of Distribution 3.X => 3.1B (tccov1001m000)

BAAN IV	Description
tdsmi0470m000	Print Selections
tdsmi1450m000	Print Letters
tdinv3420m000	Print Planned INV Purchase Orders
tcedi7401m000	Print History of Generated Messages
tdpur4401m000	Print Purchase Orders
tdpur4431m000	Print Receipts
tdsls4405m000	Print Sales Orders by Customer
tccom1401m000	Print Customers

5.4.2 BAAN Manufacturing

Conversion of Manufacturing 3.1A => BAAN IV (tccov1002m000)

BAAN IV	Description
tdpst0401m000	Print Purchase Statistics
tdpur0404m000	Print Prices by Price List and Item
tdpur1401m000	Print Inquiries
tdsst0401m000	Print Sales Statistics
tdsls0404m000	Print Prices by Price List and Item
tcmcs0408m000	Print Currency Rates

5.4.3 BAAN Finance

- Change Sign in Open Item Table (tfacp200) (tccov2001m000)
- Change Step for Antic. Unalloc /Adv Payments (tccov2002m000)
- Check Standing Order (tfcmg114) (Posting Data) (tccov2003m000)
- Check Anticipated Document (tfacp200/tfacr200) (tccov2004m000)
- Change Enumerated Field (tfcmg108) (tccov2005m000)
- Update Payment Differences from Anticipated Payment (tccov2006m000)

BAAN IV	Description
tfacp2421m000	Print Purchase Invoices
tfacr2421m000	Print Sales Invoices

5.4.4 BAAN Manufacturing

Conversion of Manufacturing 3.1 A => BAAN IV (tccov3001m000)

BAAN IV	Description
timps5410m000	Print Planned MPS Interplant Orders
timps5430m000	Print Planned MPS Purchase Orders
timrp1421m000	Print Planned MRP Purchase Orders

Conversion Units to Units by Unit Sets (tccov3002m000)
Conversion Unit Conversion Factors (tccov3003m000)

BAAN IV	Description
tiitm0401m000	Print Item Data
tiitm0410m000	Print Item Default Data
tiedm0410m000	Print Engineering Items
tipcs2421m000	Print Customized Item Data

5.5 Conversion BAAN IVa to BAAN IVb

5.5.1 BAAN Tools

After the BAAN IVb version is installed, no conversion between BAAN IVa and BAAN IVb is required for BAAN Tools.

5.5.2 BAAN Enterprise Modeler

Create a file in BAAN IVa that contains all Enterprise Modeler data by means of the Export Version Dump (tgbrg1235m000) session. All table definitions are also exported.

In BAAN IVb, use the Import Version Dump (tgbrg1245m000) session to import the created dump file.

5.5.3 BAAN Enterprise Performance Manager

Create a file in TRITON 3 or BAAN IVa, which contains all enterprise data by means of the Export Enterprise Data (tgeis1235m000) session.

In BAAN IVb, use the Import Enterprise Data (tgeis1245m000) session to import the created dump file.

5.5.4 Data conversion from BAAN IVa to BAAN IVb

After installing BAAN IVb and after a possible conversion to BAAN IVa follow the next procedure:

1. Maintain Logical Tables (ttaad4120m000). If required, the next purchase tables, tdpur040, tdpur041, tdpur045, and tdpur046 can be defined as logical tables and stored in a different company.
2. Fill the new Logistic Companies by Financial Company (tfgld000) table by means of the Maintain Logistic Companies by Financial Company (tfgld0100m000).
3. Execute the Fill Logistic Company in Open Purchase Items (tfcor0300m000) correction session. Check the **Only Test Run** option to look for any possible problems.
   This session fills the new field called Logistic Company (loco) in the Open Items (Purchase Invoices & Payments) (tfacp200) The field is the logistic company number of the order that is linked to the open entry. This session checks if an order is linked to an open entry and copied into the new field.
4. Maintain Logical Tables (ttaad4120m000). Two new tables were introduced: Customer Balances by Company (tccom011) and Supplier Balances by Company (tccom021). The new tables tccom011 and tccom021 must be logically linked in the same way as the existing tables Customers (tccom010) and Suppliers (tccom020) are logically linked.
5. Execute the Recalculate Open Entries (otfcor0260) for all financial companies, to fill in automatically the invoice balance of the suppliers.
6. Execute the Recalculate Open Entries (otfcor0261) for all financial companies, to fill in automatically the invoice balance of the customers.
7. Execute the Rebuild Customer Order Balance (tccom2210m000), to fill in automatically all customer order balances.
8. Execute the Rebuild Supplier Order Balance (tccom2220m000), to fill in automatically all customer order balances.

5.6 Converting localizations to BAAN IV

The sections below describe the conversion programs for the localizations.

5.6.1 Proforma Invoice Conversion from 3.1*x* to BAAN IV*x* (tfcmgl900m000)

The conversion program tfcmgl900m000 converts the Proforma Invoice tables present in 3.1*x*GLO to the standard BAAN IV*x* tables. This utility is optional. The Proforma Invoice feature was moved into the standard for BAAN IVa and as such all table names were changed. This conversion program is executed from the command line for those customers using the Proforma Invoice feature.

5.6.2 EDI 'Message Generated' Conversion from BAAN IVa to BAAN IVb (tccovl999m000)

The utility tccovl999m000 is optional. This utility sets the EDI 'Message Generated' flag on Sales and Purchase Orders.

5.7 Removing the 3.*x* software after the conversion

After the transition to BAAN IVb has been successfully completed, the TRITON 3.*x* software (including package VRCs, package combinations, custimizations, and localizations) must be removed.

6 Appendix A: Upgrading customizations

This chapter describes the steps needed to upgrade both external and in-house customizations. This applies especially to between TRITON 3 to BAAN IV. For small differences, between BAAN IVa and BAAN IVb, see also the online Help of the Upgrade Customizations to New Standard method in BAAN Tools.

6.1 Recommendations

After a new version/release has been installed by means of the installation program, this new software is not yet active. To activate this software, a new package combination can be created to which a company can be linked to test the software. In the meantime, the remaining companies continue using the old TRITON 3.1 software. After the software has been tested and approved, these remaining companies are also migrated to the new software.

A distinction is made between migrating customizations developed by Baan or a Baan dealer and customizations developed by the customer (so-called in-house customizations). In this context, customizations are modifications to the standard software.

6.2 Print differences between two PVRCs

In a single environment scenario you can print all the differences by means of the Compare Package VRCs (ttadv6450m000) session, You can either print global change information or detailed information by selecting the Detailed Information (excl. Technical Documentation) check box.

6.3 Upgrade of in-house customizations

To upgrade in-house customizations, take the following steps:

1. The tdB40O_a_cu01 and tdB40O_a_cu02 PVRCs must be created.
2. The directories for software components must be set for both PVRCs.
3. The software components must be exported from the TRITON 3.1 environment by means of the Export Data Dictionary (ttiex1280m000) session and imported in the BAAN IV environment by means of the Import Data Dictionary (ttiex1285m000) session, namely:
   - from td3.1O_b_cu01 to tdB40O_a_cu01
   - from td3.1O_b_cu02 to tdB40O_a_cu02
4. The VRC tree must be defined.

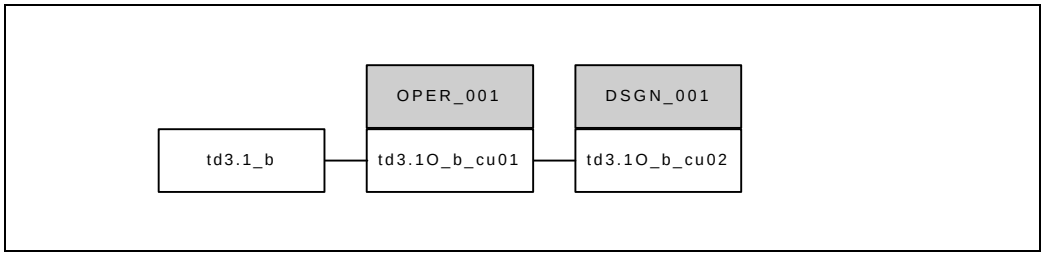

Figure 6-1 Start position

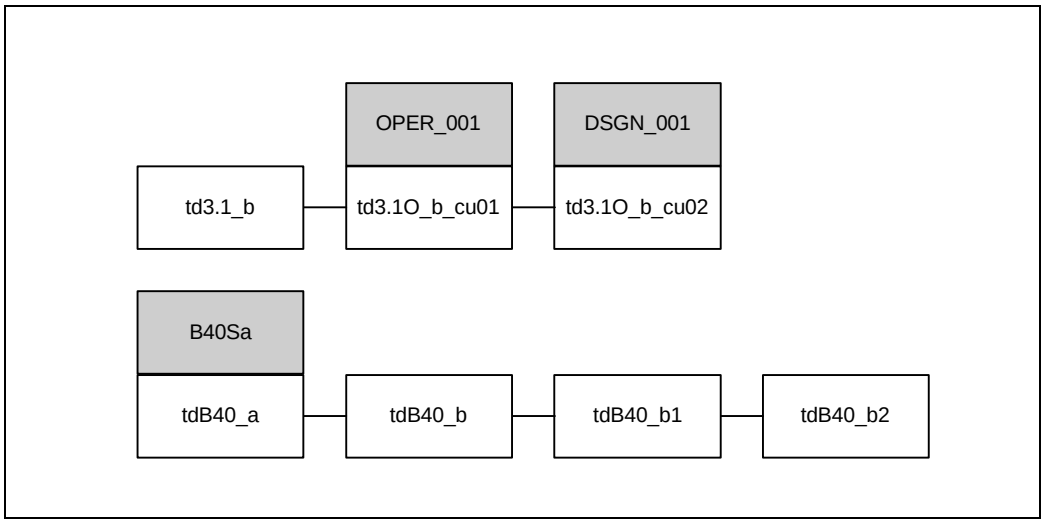

Figure 6-2 Testing the new version

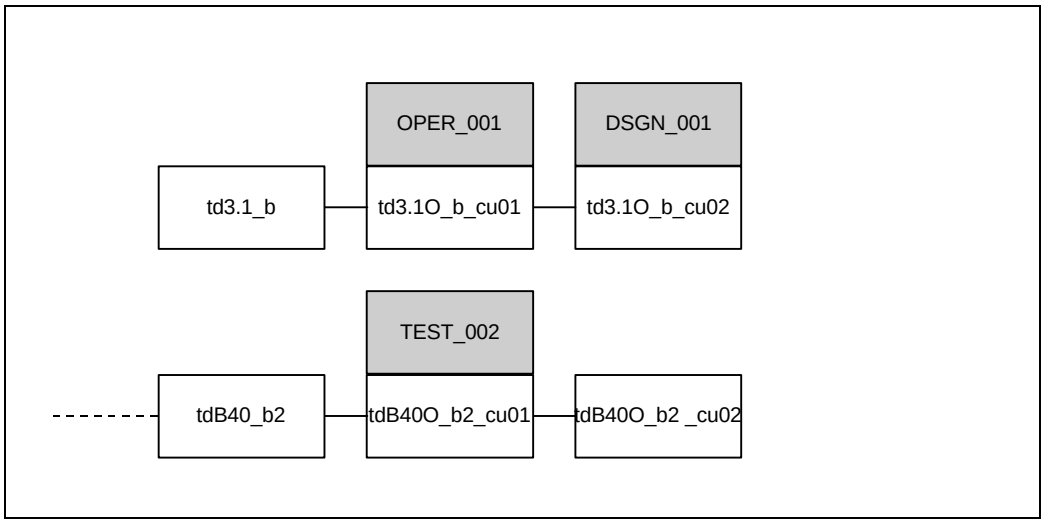

Figure 6-3 Testing the in-house customization

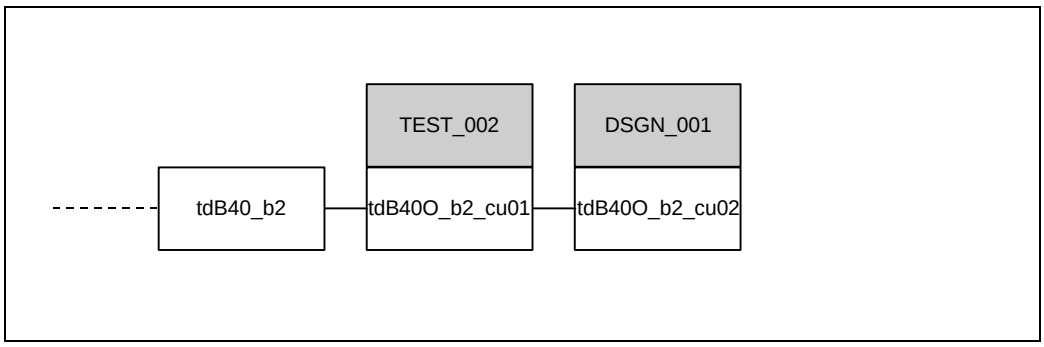

Figure 6-4 Released update

6.4 Upgrading customized software and in-house customizations

To upgrade both customized software and in-house customizations, the same procedure can be followed as for in-house customizations only. The only difference is that this procedure is extended with a step in which the external customizations are tested.

An update of the standard involves an update of the customization which is derived from that standard. This is shown in Figures 6-5 through 6-9.

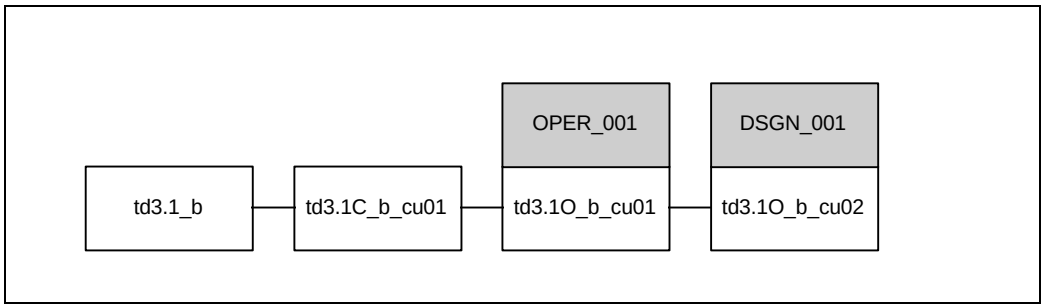

Figure 6-5 Start position

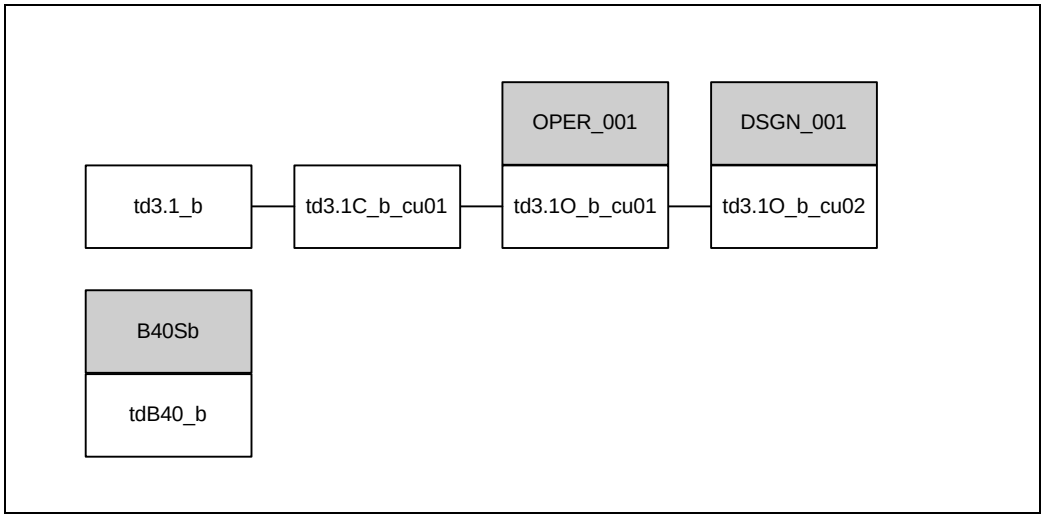

Figure 6-6 Testing the standard installation of BAAN IV

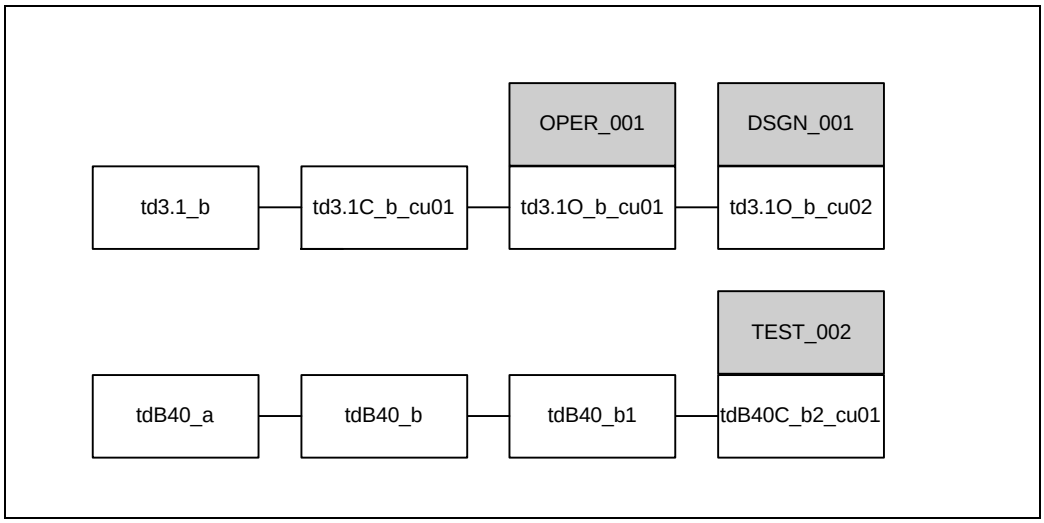

Figure 6-7 Testing the external customization

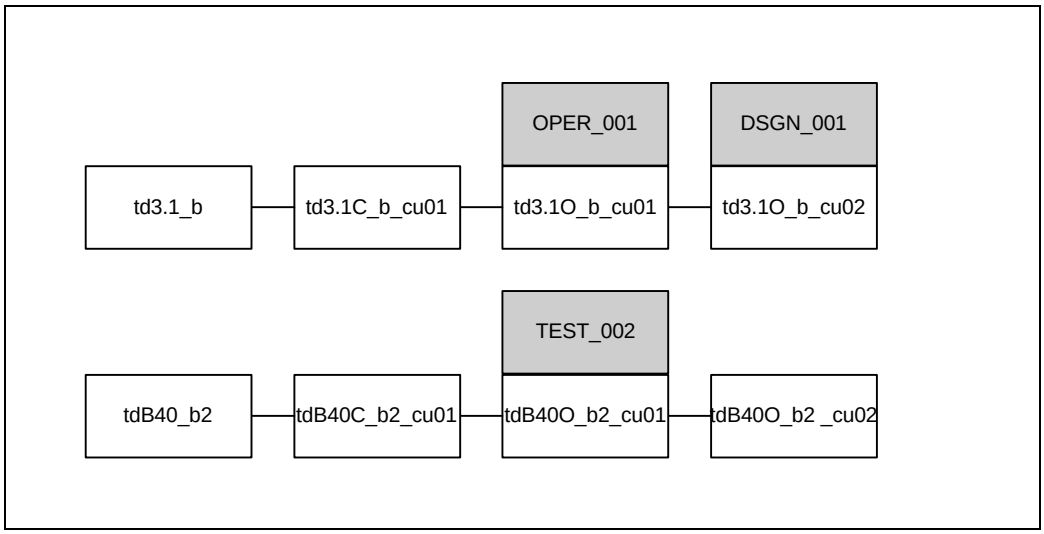

Figure 6-8 Testing the in-house customization

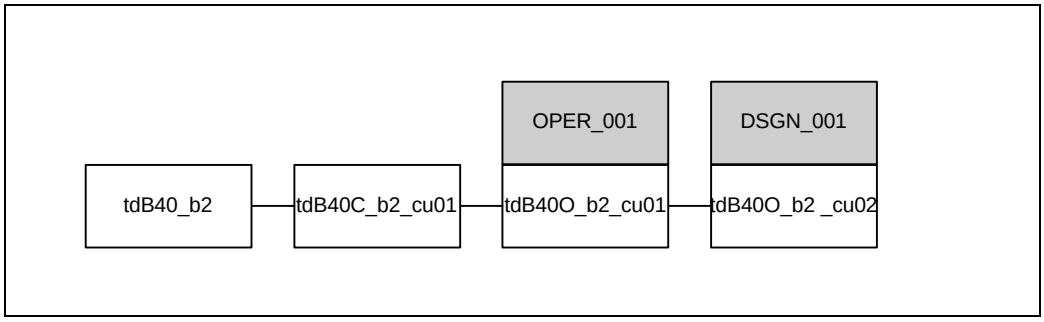

Figure 6-9 Released software

Prior to converting the data, the data model in BAAN IV must be assessed as being the correct data model and the correction program (tccov0000m000) has to be adapted to this.

7 Appendix B: Troubleshooting

This appendix contains a listing of messages and errors you may encounter during the conversion of TRITON 3 to BAAN IV.

7.1 Conversion of Distribution 3.1 to BAAN IV (tccov1001m000)

This section lists the messages that can be displayed during the conversion of Distribution 3.1 to BAAN IV (tccov1001m000).

7.1.1 Message 1: START OF CONVERSION tccov1001m000 by user <*logon name*>

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.1.2 Message 2: END OF CONVERSION tccov1001m000 by user <*logon name*>

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.1.3 Message 3: 'Letter layout: <tdsmi150.clet> converted'

This message informs you that letter layouts in tdsmi150 are converted. For each letter layout in tdsmi150 is recorded if it has been converted.

Type of message

Information.

Possible actions

No action required.

7.1.4 Message 4: 'Selection expression: <tdsmi070.cnsc> converted'

This message informs you that selection expressions in tdsmi070 are converted. For each selection expression in tdsmi070 is recorded if it has been converted.

Type of message

Information.

Possible actions

No action required.

7.1.5 Message 5: 'Recommended Purchase Order: <tdinv310.orno> converted'

This message informs you that the company number in tdinv310 is filled with supplier's company. For each recommended Purchase Order in tdinv310 it is recorded if it has been converted.

Type of message

Information.

Possible actions

No action required.

7.1.6 Message 6: 'Outgoing Message: <tcedi700.mess> converted'

This message informs you that the parent relation field in tcedi700 is filled. For each outgoing message in tcedi700 it is recorded if it has been converted.

Type of message

Information.

Possible actions

No action required.

7.1.7 Message 7: 'Batch Number: <tcedi720.bano> converted'

This message informs you that tcedi720 is filled with values from tcedi701. For each batch number in tcedi720 it is recorded if it has been converted.

Type of message

Information.

Possible actions

No action required.

7.1.8 Message 8: 'Company Purchase Order: <tdpur045.orno> <tdpur045.pono> <tdpur045.srnb> converted'

This message informs you that the financial company in tdpur045 is updated. For each purchase order in tdpur045 it is recorded if the financial company has been updated.

Type of message

Information.

Possible actions

No action required.

7.1.9 Message 9: 'Financial Customer Group Sales Order: <tdsls040.orno> converted'

This message informs you that the financial customer group in tdsls040 is updated. For each sales order in tdsls040 it is recorded if the Financial Customer Group has been updated.

Type of message

Information.

Possible actions

No action required.

7.2 Conversion of Distribution 3.1 to BAAN IV (tccov1002m000)

This section lists the messages that can be displayed during the conversion of Distribution 3.1 to BAAN IV (tccov1002m000). The messages are stored in the log file **$BSE/log/log.cov1002**.

Note: The messages 3 up to 18 can only be printed if the 'Convert Statistics' parameter is set to Yes. The purpose of all these conversions is to convert all sort strings in the statistics tables to a new domain.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full Description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.2.1 Message 1: START OF CONVERSION tccov1002m000 by user *<logon name>*

This message informs you about the start date and time of the session by the specified user.

Type of message Information.

Possible actions No action required.

7.2.2 Message 2: END OF CONVERSION tccov1002m000 by user *<logon name>*

This message informs you about the end date and time of the session by the specified user.

Type of message Information.

Possible actions No action required.

7.2.3 Message 3: 'Conversion error: sst010.sstr:<sst010.sstr>'

This message informs you that the table tdsst010 has a physical database problem, for example, it is corrupted.

Type of message Error.

Possible actions Use the key value printed in the error message to check the table manually.

7.2.4 Message 4: 'Conversion error: sst015.sstr: <sst015.sstr>'

This message informs you that the table tdsst015 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.5 Message 5: 'Conversion error: sst018.sstr: <sst018.sstr>'

This message informs you that the table tdsst018 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.6 Message 6: 'Conversion error: sst020.sstr: <sst020.str>'

This message informs you that the table tdsst020 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.7 Message 7: 'Conversion error: sst025.sstr: <sst025.str>'

This message informs you that the table tdsst025 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.8 Message 8: 'Conversion error: sst028.sstr: <sst028.str>'

This message informs you that the table tdsst028 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.9 Message 9: 'Conversion error: sst050.sstr: <sst050.str>'

This message informs you that the table tdsst050 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.10 Message 10: 'Conversion error: sst030.sstr: <sst030.str>'

This message informs you that the table tdsst030 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.11 Message 11: 'Conversion error: pst010.sstr:<pst010.sstr>'

This message informs you that the table tdpst010 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.12 Message 12: 'Conversion error: pst015.sstr:<pst015.sstr>'

This message informs you that the table tdpst015 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.13 Message 13: 'Conversion error: pst018.sstr:<pst018.sstr>'

This message informs you that the table tdpst018 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.14 Message 14: 'Conversion error: pst020.sstr:<pst020.sstr>'

This message informs you that the table tdpst020 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.15 Message 15: 'Conversion error: pst025.sstr:<pst025.sstr>'

This message informs you that the table tdpst025 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.16 Message 16: 'Conversion error: pst028.sstr:<pst028.sstr>'

This message informs you that the table tdpst028 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.17 Message 17: 'Conversion error: pst050.sstr:<pst050.sstr>'

This message informs you that the table tdpst050 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.18 Message 18: 'Conversion error: pst030.sstr:<pst050.sstr>'

This message informs you that the table tdpst030 has a physical database problem, for example, it is corrupted.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.19 Message 19: 'Conversion error: mcs047.ckon: <mcs047.ckon> mcs047.grno:<mcs047.grno>'

This message informs you that the table tcmcs047 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to update the first free numbers for proforma sales invoices.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.20 Message 20: 'Conversion error: edi005.mess <tcedi005.mess>'

This message informs you that the table tcedi005 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to update the names of libraries used within EDI.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.21 Message 21: 'Conversion error: pur002.opol <pur002.qono> <pur002.pono>'

This message informs you that the table tdpur002 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to update the order policy field in table pur002.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.22 Message 22: 'Conversion error: pur011.opol <pur011.qono> <pur011.pono>'

This message informs you that the table tdpur011 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to update the order policy field in table pur011

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.23 Message 23: 'Conversion error: pur034.suno <pur034.suno> pur034.cpgp <pur034.cpgp> pur034.dile <pur034.dile>'

This message informs you that the table tdpur034 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in pur034.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.24 Message 24: 'Conversion error: pur035.cplp <pur035.cplp> pur035.cpgp <pur035.cpgp> pur035.dile <pur035.dile>'

This message informs you that the table tdpur035 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in pur035.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.25 Message 25: 'Conversion error: pur036.suno <pur036.suno> pur036.item <pur036.item> pur036.cntr <pur036.cntr> pur036.dile <pur036.dile>'

This message informs you that the table tdpur036 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in pur036.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.26 Message 26: 'Conversion error: pur037.cplp <pur037.cplp> pur037.item <pur037.item> pur037.cntr <pur037.cntr> pur037.dile <pur037.dile>'

This message informs you that the table tdpur037 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in pur037.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.27 Message 27: 'Conversion error: inv300.orno <tdinv300.orno>'

This message informs you that the table tdinv300 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the planner field in inv300.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.28 Message 28: 'Conversion error: inv310.orno <tdinv310.orno>'

This message informs you that the table tdinv310 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the planner field in inv310.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.29 Message 29: 'Conversion error: sls002.opol <sls002.qono> <sls002.pono>'

This message informs you that the table tdsls002 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the order policy field in sls002.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.30 Message 30: 'Conversion error: sls011.opol <sls011.qono> <sls011.pono>'

This message informs you that the table tdsls011 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the order policy field in sls011.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.31 Message 31: 'Conversion error: sls034.cuno <sls034.cuno> sls034.cpgp <sls034.cpgp> sls034.dile <sls034.dile>'

This message informs you that the table tdsls034 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill unit field in sls034.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.32 Message 32: 'Conversion error: sls035.cpls <sls035.cpls> sls035.cpgp <sls035.cpgp> sls035.dile <sls035.dile>'

This message informs you that the table tdsls035 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in sls035.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.33 Message 33: 'Conversion error: sls036.cuno <sls036.cuno> sls036.item <sls036.item> sls036.cntr <sls036.cntr> sls036.dile <sls036.dile>'

This message informs you that the table tdsls036 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in sls036.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.34 Message 34: 'Conversion error: sls037.cpls <sls037.cpls> sls037.item <sls037.item> sls037.cntr <sls037.cntr> sls037.dile <sls037.dile>'

This message informs you that the table tdsls037 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the unit field in sls037.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.35 Message 35: 'Conversion error: sls045.orno <sls045.orno> sls045.pono <sls045.pono> sls045.srnb <sls045.srnb>'

This message informs you that the table tdsls045 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the procedure mask, rate, and rate factor in sls045.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.36 Message 36: 'Conversion error: sls051.orno <sls051.orno> sls051.trdt <sls051.trdt> sls051.trtm <sls051.trtm>'

This message informs you that the table tdsls051 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the procedure mask, rate, and rate factor in sls051.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.37 Message 37: 'Conversion error: com999.sern <tccom999.sern>'

This message informs you that the table tccom999 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill com999 from table mcs095.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.38 Message 38: 'Conversion error: mcs042.blcs: blocking stepno not in procedure'

In TRITON 3 there was one system parameter in which the order step was defined at which sales orders must be blocked. In BAAN IV the blocking step is not defined at system level but at order type level. The blocking step must be an existing step in the order procedure. Due to the fact that in Triton3 it was a system parameter, it is very well possible that the blocking step is not an existing step in the concerning order procedure of the order type. In this case the blocking step at order type level is left empty.

Type of message

Error.

Possible actions

No action required during conversion. During conversion no action. After conversion you must manually check and fill the blocking step of the order types mentioned in the error report.

7.2.39 Message 39: 'Conversion error: tcmcs042.pmsk: <tcmcs042.cotp> <tcmcs042.pmsk>'

This message informs you that the table table tcmcs042 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill procedure mask in mcs042.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.40 Message 40: 'Conversion error: tdrpl004.pmsk: <tdrpl004.cotp> <tdrpl004.pmsk>'

This message informs you that the table table tdrpl004 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to extend the procedure mask field in rpl004.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.41 Message 41: 'Conversion error: tdpur045.pmsk: <tdpur045.orno> <tdpur045.pono> <tdpur045.srnb> <tdpur045.pmsk>'

This message informs you that the table table tdrpl004 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to extend the procedure mask field in pur045.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.2.42 Message 42: 'Conversion error: tdrpl100.pmsk: <tdrpl100.orno> <tdrpl100.pono> <tdrpl100.srnb> <tdrpl100.pmsk>'

This message informs you that the table table tdrpl100 has a physical database problem, for example, it is corrupted. The purpose of this conversion is to fill the procedure mask in rpl100.

Type of message

Error.

Possible actions

Use the key value printed in the error message to check the table manually.

7.3 Changing a sign in open item table (tccov2001m000)

This section lists the messages that can be displayed during the execution of tccov2001m000. The messages are stored in the log file **$BSE/log/log.cov2001**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full Description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.3.1 Message 1: START OF CONVERSION tccov2001m000 by user <*login name*>

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.3.2 Message 2: Convert CDAM error: <supplier>,<transactiontype>,<document>,<linenumber>, <transactiontype>,<document>,<linenumber>, <currency difference>

For each changed sign in tfacp200 the information is logged.

Type of message

Information.

Possible actions

No action required.

7.3.3 Message 3: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.4 Change step for anticipated unallocated/advance payments (tccov2002m000)

This section lists the messages that can be displayed during the execution of tccov2002m000. The messages are stored in the log file **$BSE/log/log.cov2002**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.4.1 Message 1: START OF CONVERSION tccov2002m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.4.2 Message 2: tfacp200: <supplier>,<transactiontype>,<document>,<linenumber>, <transactiontype>,<document>,<linenumber>

For each filled step in tfacp200 the key information is logged.

Type of message

Information.

Possible actions

No action required.

7.4.3 Message 3: tfacr200: <customer>,<transactiontype>,<document>, <linenumber>,<transactiontype>,<document>, <linenumber>

For each filled step in tfacr200 the key information is logged.

Type of message

Information.

Possible actions

No action required.

7.4.4 Message 4: END OF CONVERSION

This message informs you about the end date and time of session.

Type of message

Information.

Possible actions

No action required.

7.5 tccov2003m000: Checking standing order (tfcmg114)

This section lists the messages that can be displayed during the execution of tccov2003m000. The messages are stored in the log file **$BSE/log/log.cov2003**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.5.1 Message 1: START OF CONVERSION tccov2003m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.5.2 Message 2: Posting Data: <bank>,<document>,<company>, <transactiontype>,<document>,<supplier>, <standing order>,<serial number>

For each changed standing order in tfcmg114 the information is logged.

Type of message

Information.

Possible actions

No action required.

7.5.3 END OF CONVERSION

This message informs you about the end date and time of session.

Type of message

Information.

Possible actions

No action required.

7.6 tccov2004m000: Changing anticipated documents in tfacp200/tfacr200/tfgld102/tfgld106

This section lists the messages that can be displayed during the execution of tccov2004m000. The messages are stored in the log file **$BSE/log/log.cov2004**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.6.1 Message 1: START OF CONVERSION tccov2004m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.6.2 Message 2: tfacp200: <transactiontype>,<document>, <anticipated transactiontype>,<anticipated document>

For each blanked anticipated document in tfacp200 information is logged.

Type of message

Information.

Possible actions

No action required.

7.6.3 Message 3: tfacr200: <transactiontype>,<document>, <anticipated transactiontype>,<anticipated document>

For each blanked anticipated document in tfacr200 information is logged.

Type of message

Information.

Possible actions

No action required.

7.6.4 Message 4: tfgld102: <fiscal year>, <batchnumber>, <transactiontype>, <document>, <correction transactiontype>,<correction document>

For each blanked correction document in tfgld102 information is logged.

Type of message

Information.

Possible actions

No action required.

7.6.5 Message 5: tfgld106: <transactiontype>,<document>, <correction transactiontype>,<correction invoice>

For each blanked correction document in tfgld106 information is logged.

Type of message

Information.

Possible actions

No action required.

7.6.6 Message 6: END OF CONVERSION

This message informs you about the end date and time of session.

Type of message

Information.

Possible actions

No action required.

7.7 tccov2005m000: Changing enumerated field (tfcmg108)

This section lists the messages that can be displayed during the execution of tccov2005m000. The messages are stored in the log file **$BSE/log/log.cov2005**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.7.1 Message 1: START OF CONVERSION tccov2005m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.7.2 Message 2: tfcmg108: <payment batch>, <original company>, <payment type>, <supplier>,<order>

For each changed transaction in tfcmg108 information is logged.

Type of message

Information.

Possible actions

No action required.

7.7.3 Message 3: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.8 tccov2006m000: Updating payment difference for anticipated document

This section lists the messages that can be displayed during the execution of tccov2006m000. The messages are stored in the log file **$BSE/log/log.cov2006**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.8.1 Message 1: START OF CONVERSION tccov2006m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.8.2 Message 2: tfacp200 <supplier>,<transactiontype>,<document>,<linenumber>, <transactiontype>,<document>,<linenumber>, <payment difference>

For each filled step in tfacp200 the key information is logged.

Type of message

Information.

Possible actions

No action required.

7.8.3 Message 3: tfacr200: <customer>,<transactiontype>,<document>, <linenumber>,<transactiontype>,<document>, <linenumber>, <payment difference>

For each filled step in tfacr200 the key information is logged.

Type of message

Information.

Possible actions

No action required.

7.8.4 Message 1: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.9 tccov3001m000: Converting Manufacturing 3.1 to BAAN IV

This section lists the messages that can be displayed during the execution of tccov3001m000. The messages are stored in the log file **$BSE/log/log.cov3001**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full Description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.9.1 Message 1: START OF CONVERSION tccov3001m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.9.2 Message 2: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.10 tccov3002m000: Conversion Units to Units by Unit Sets

This section lists the messages that can be displayed during the execution of tccov3002m000. The messages are stored in the log file **$BSE/log/log.cov3002**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full Description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.10.1 Message 1: START OF CONVERSION tccov3002m000 by user *<login name>*

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.10.2 Message 2: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.10.3 Message 3: tiitm001:<item.code>.<unit.set>.<units>

For each converted item in tiitm001 the unit set with its units will be logged.

Type of message

Information.

Possible actions

No action required.

7.10.4 Message 4: tiitm002:<item.group>.<unit.set>.<units>

For each converted item default in tiitm002 the unit set with its units will be logged.

Type of message

Information.

Possible actions

No action required.

7.10.5 Message 5: tipcs021:<project.code>.<item.code>.<unit.set>.<units>

For each converted project item in tipcs021 the unit set with its units will be logged.

Type of message

Information.

Possible actions

No action required.

7.10.6 Message 6: tipcs002:<budget.code>.<calculation.part>.<unit.set>.<units>

For each converted calculation part in tipcs002 the unit set with its units will be logged.

Type of message

Information.

Possible actions

No action required.

7.10.7 Message 7: tiedm010:<enginering.item>.<unit.set>.<units>

For each enginering item in tiedm010 the unit set with its units will be logged.

Type of message

Information.

Possible actions

No action required.

7.10.8 Message 8: tppdm025:<equipment.code>.<unit.set>.copied.from.itm001.<unit>

Message will be logged when unit of equipment in tppdm025 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.9 Message 9: tppdm025:<equipment.code>.Error:unit.not.in.unit.set.of.itm001

Error message will be logged when there is an integration to TRITON Distribution and unit of equipment in tppdm025 is NOT found in the already generated unit set.

Type of message

Error.

Possible actions

Add unit of equipment to a unit set and run this program again.

7.10.10 Message 10: tppdm025:<equipment.code>.<unit.set>.<units>

Message will be logged for each equipment in tppdm025 when there is NO integration to TRITON Distribution and unit is not found in an already existing unit set.

Type of message

Information.

Possible actions

No action required.

7.10.11 Message 11: tppdm035<subcontracting.code>.<unit.set>.copied.from.itm001.<unit>

Message will be logged when unit of subcontracting tppdm035 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.12 Message 12: tppdm035<subcontracting.code>.<unit.set>.copied.from.itm001.<unit>

Error message will be logged when there is an integration to TRITON Distribution and unit of subcontracting in tppdm035 is NOT found in the already generated unit set.

Type of message

Error.

Possible actions

Add unit of subcontracting to a unit set and run this program again.

7.10.13 Message 13: tppdm035:<subcontracting.code>.<unit.set>.<units>

Message will be logged for each subcontracting in tppdm035 when there is NO integration to TRITON Distribution and unit is not found in an already existing unit set.

Type of message

Information.

Possible actions

No action required.

7.10.14 Message 14: tppdm625:<project>.<equipment.code>.<unit.set>.copied.from.pcs021.<unit>

Message will be logged when unit of project equipment in tppdm625 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.15 Message 15: tppdm625:<project>.<equipment.code>.Error:unit.not.in.unit.set.of.pcs021

Error message will be logged when there is an integration to TRITON Distribution and unit of project equipment in tppdm625 is NOT found in the already generated unit set.

Type of message

Error.

Possible actions

Add unit of project equipment to a unit set and run this program again.

7.10.16 Message 16: tppdm625:<project>.<equipment.code>.<unit.set>.<units>

Message will be logged for each project equipment in tppdm625 when there is NO integration to TRITON Distribution and unit is not found in an already existing unit set.

Type of message

Information.

Possible actions

No action required.

7.10.17 Message 17: tppdm635:<project>.<subcontracting.code>.<unit.set>.copied.from.pcs021.<unit>

Message will be logged when unit of project subcontracting in tppdm635 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.18 Message 18: tppdm635:<project>.<subcontracting.code>.Error:unit.not.in.unit.set.of.pcs021

Error message will be logged when there is an integration to TRITON Distribution and unit of project subcontracting in tppdm635 is NOT found in the already generated unit set.

Type of message

Error.

Possible actions

Add unit of project subcontracting to a unit set and run this program again.

7.10.19 Message 19: tppdm635:<project>.<subcontracting.code>.<unit.set>.<units>

Message will be logged for each project subcontracting in tppdm635 when there is NO integration to TRITON Distribution and unit is not found in an already existing unit set.

Type of message

Information.

Possible actions

No action required.

7.10.20 Message 20: tppdm605:<project>.<project.item>.Error:item.not.in.pdm625

Error message will be logged when equipment of a project item in tppdm605 is not found in tppdm625.

Type of message

Error.

Possible actions

Add equipment with the currect unit to tppdm625 and run this program again.

7.10.21 Message 21: tppdm605:<project>.<project.item>.Error:item.not.in.pdm635

Error message will be logged when subcontracting of a project item in tppdm605 is not found in tppdm635.

Type of message

Error.

Possible actions

Add subcontracting with the currect unit to tppdm635 and run this program again.

7.10.22 Message 22: tppdm605:<project>.<itemcode>.<unit.set>.<units>

Message will be logged when units of standard project item in tppdm605 is found in an existing unit set.

Type of message

Information.

Possible actions

No action required.

7.10.23 Message 23: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pcs021

Error message will be logged when when there is an integration to TRITON Distribution and unit(s) of a standard project item in tppdm605 is not found in existing unitsets.

Type of message

Error.

Possible actions

Add unit(s) of standard project item to a unit set and run this program again.

7.10.24 Message 24: tppdm605:<project>.<itemcode>.<unit.set>.copied.from.pdm625

Message will be logged when unit(s) of project equipment item in tppdm605 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.25 Message 25: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pdm625

Error message will be logged when unit(s) of a project equipment titem in tppdm605 is not found in existing unitsets.

Type of message

Error.

Possible actions

Add unit(s) of project equipment item to a unit set and run this program again.

7.10.26 Message 26: tppdm605:<project>.<itemcode>.<unit.set>.copied.from.pdm635

Message will be logged when unit(s) of project subcontracting item in tppdm605 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.27 Message 27: tppdm605:<project>.<itemcode>.Error:unit.not.in.unit.set.of.pdm635

Error message will be logged when unit(s) of a project subcontracting titem in tppdm605 is not found in existing unitsets.

Type of message

Error.

Possible actions

Add unit(s) of project subcontracting item to a unit set and run this program again.

7.10.28 Message 28: tppdm604:<project>.<itemcode>.<unitset>.copied.from.tppdm605

Message will be logged when alternate unit of a project item in tppdm604 is found in the already generated unit set.

Type of message

Information.

Possible actions

No action required.

7.10.29 Message 29: tppdm604:<project>.<itemcode>.<unitset>.Error:unit.not.in.unit.set.of.pdm605

Error message will be logged when alternate unit of a project item in tppdm604 is not found in the already generated unit set.

Type of message

Error.

Possible actions

Add unit to a unit set and run this program again.

7.10.30 Message 30: Base Weight Unit not filled

Error message will be logged when Base Length Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Length Unit in tcmcs000 and run the program again.

7.10.31 Message 31: Base Length Unit not filled

Error message will be logged when Base Length Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Length Unit in tcmcs000 and run the program again.

7.10.32 Message 32: Base Area Unit not filled

Error message will be logged when Base Area Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Area Unit in tcmcs000 and run the program again.

7.10.33 Message 33: Base Volume Unit not filled

Error message will be logged when Base Volume Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Volume Unit in tcmcs000 and run the program again.

7.10.34 Message 34: Base Time Unit not filled

Error message will be logged when parameters are not present in tcmcs000.

Type of message

Fatal error.

Possible actions

Enter Time Unit in tcmcs000 and run the program again.

7.10.35 Message 35: parameters MCS not present

Error message will be logged when parameters are not present in tcmcs000.

Type of message

Error.

Possible actions

Enter parameters in tcmcs000.

7.10.36 Message 36: physical quantity '<description>' has duplicates (conversion factor = 1.0)

Error message will be logged when duplicate units with conversion factor = 1.0 and same physical quantity are found in tcmcs001. The program will abort.

Type of message

Fatal error.

Possible actions

Enter units for all physical quantities to tcmcs001.

7.10.37 Message 37: No units present

Error message will be logged when duplicate units with conversion factor = 1.0 and same physical quantity are found in tcmcs001. The program will abort.

Type of message

Fatal error.

Possible actions

Enter units for all physical quantities to tcmcs001.

7.10.38 Message 38: base unit physical quantity '<description>' not known (conversion factor = 1.0)

Error message will be logged when no unit is found with a certain physical quantity and conversion factor = 1.0. The program will abort.

Type of message

Fatal error.

Possible actions

Enter unit for that physical quantity to tcmcs001.

7.10.39 Message 39: domain <domain.name>has no enumerated values

This error will be logged when a certain domain has not enumerated values. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor or enter enumerated values to enum domain.

7.10.40 Message 40: Error <error.code> in qss.search(<search.mode>,<search.argument>,<table>,<definition>,<number>)

This will be a program error. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor.

7.10.41 Message 41: Error <error.code> in qss.sort(<table>,<definition>,<number>)

This will be a program error. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor.

7.11 tccov3003m000: Conversion Unit Conversion Factors

This section lists the messages that can be displayed during the execution of tccov3003m000. The messages are stored in the log file **$BSE/log/log.cov3003**.

The messages are stored as follows:

- Type of output : Only process information will be logged into this file.
- Full description : Start and end of process will be logged.
- Format of output : MM/DD/YY-mm:ss:%s

MM	= Month
DD	= Day
YY	= Year
mm	= Minutes
ss	= Seconds
%s	= <Message>

7.11.1 Message 1: START OF CONVERSION tccov3003m000 by user <*login name*>

This message informs you about the start date and time of the session by the specified user.

Type of message

Information.

Possible actions

No action required.

7.11.2 Message 2: END OF CONVERSION

This message informs you about the end date and time of the session.

Type of message

Information.

Possible actions

No action required.

7.11.3 Message 3: CONFLICT: item ><item.code>><item.unit>><alternate.unit>><conversion.factor>> conflict general conversion

A message will be logged when a conversion factor is found for an item whose item unit has the same physical quantity (not pieces) as the physical quantity of an alternate unit.

Type of message

Warning.

Possible actions

Change conversion factors afterwards (optional).

7.11.4 Message 4: CONFLICT proj/item ><project>><item.code>><unit>><purchase.unit>><conversion.factor>> conflict general conversion

A message will be logged when a conversion factor is found for an project item whose item unit has the same physical quantity (not pieces) as the physical quantity of the purchase unit of that project item.

Type of message

Warning.

Possible actions

Change conversion factors afterwards (optional).

7.11.5 Message 5: CONFLICT proj/item ><project>><item.code>><unit>><purchase.price.unit>><conversion.factor>> conflict general conversion

A message will be logged when a conversion factor is found for an project item whose item unit has the same physical quantity (not pieces) as the physical quantity of the purchase price unit of that project item.

Type of message

Warning.

Possible actions

Change conversion factors afterwards (optional).

7.11.6 Message 6: CONFLICT proj/item ><project>><item.code>><unit>><storage.unit>><conversion.factor>> conflict general conversion

A message will be logged when a conversion factor is found for an project item whose item unit has the same physical quantity (not pieces) as the physical quantity of the storage unit of that project item.

Type of message

Warning.

Possible actions

Change conversion factors afterwards (optional).

7.11.7 Message 7: tiitm004:<project>.<item.code>.<item.group>.<base.unit>.<unit>.<conversion.factor>

This message will be logged for each written general conversion factor.

Type of message

Information.

Possible actions

No actions required.

7.11.8 Message 8: Base Weight Unit not filled

Error message will be logged when Base Weight Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Weight Unit in tcmcs000 and run program again.

7.11.9 Message 9: Base Length Unit not filled

Error message will be logged when Base Length Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Length Unit in tcmcs000 and run program again.

7.11.10 Message 10: Base Area Unit not filled

Error message will be logged when Base Area Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Area Unit in tcmcs000 and run program again.

7.11.11 Message 11: Base Volume Unit not filled

Error message will be logged when Base Volume Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Volume Unit in tcmcs000 and run the program again.

7.11.12 Message 12: Base Time Unit not filled

Error message will be logged when Base Time Unit is not filled in tcmcs000. The program will abort.

Type of message

Fatal error.

Possible actions

Enter Time Unit in tcmcs000 and run the program again.

7.11.13 Message 13: parameters MCS not present

Error message will be logged when parameters are not present in tcmcs000.

Type of message

Error.

Possible actions

Enter parameters in tcmcs000.

7.11.14 Message 14: physical quantity '<description>' has duplicates (conversion factor = 1.0)

Error message will be logged when duplicate units with conversion factor = 1.0 and same physical quantity are found in tcmcs001. The program will abort.

Type of message

Fatal error.

Possible actions

Remove unused units from tcmcs001 or change conversion factors of duplicate units with same physical quantity.

7.11.15 Message 15: No units present

Error message will be logged when no units are found in tcmcs001. The program will abort.

Type of message

Fatal error.

Possible actions

Enter units for all physical quantities to tcmcs001.

7.11.16 Message 16: base unit physical quantity '<description>‘ not known (conversion factor = 1.0)

Error message will be logged when no unit is found with a certain physical quantity and conversion factor = 1.0. The program will abort.

Type of message

Fatal error.

Possible actions

Enter unit for that physical quantity to tcmcs001.

7.11.17 Message 17: domain <domain.name>has no enumerated values

This error will be logged when a certain domain has not enumerated values. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor or enter enumerated values to enum domain.

7.11.18 Message 18: Error <error.code> in qss.search(<search.mode>,<search.argument>,<table>,<definition>,<number>)

This will be a program error. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor.

7.11.19 Message 19: Error <error.code> in qss.sort(<table>,<definition>,<number>)

This will be a program error. The program will abort.

Type of message

Fatal error.

Possible actions

Ask for support from software vendor.

7.11.20 Message 20: END OF CONVERSION; <number.conflicts> CONFLICTS FOUND

This message will be logged when conflicts have been registrated.

Type of message

Information.

Possible actions

No actions required.

7.11.21 Message 21: END OF CONVERSION; NO CONFLICTS

This message will be logged when no conflicts have been registrated.

Type of message

Information.

Possible actions

No actions required.